DRAGON USER

The independent Dragon magazine

January 1989

Contents

Brian Cadge apologises for missing this month's copy deadline while his computer was at the mender's.

Editorial

OH, drat, no, sorry, make that damn. What do I say now? I thought about crawling under my desk and staying there, pretending that I never heard nuffing till later, but as Bob has provided a formal statement about Dragon User's future (or lack of it — see page 5), I can't get away with that.

"Are you going to the Weston Show", I said to him on Thursday, uneasily because work was piling up and I didn't see getting there.

"I don't really feel like it at the moment", he said unexpectedly, and then explained that subscriptions had dropped sharply, and he couldn't see how we could continue beyond the current issue.

I don't have time to remove all the 'next months' from the columns, but it looks as though Dragon User will die with this edition.

I am going to miss the old beast. I won't starve this week. But I don't like thinking of Dragon resources wasted. Something may be worked out so that our material will be available from another source. Support your remaining editors — Paul Grade, Simon Jones, Andrew Hill, Donald Morrison. They are still The Ones Who Do The Work.

Telephone number
(01) 570 8335

Editor
HELEN ARMSTRONG

Production Editor
HELEN ARMSTRONG/ARTSET

Administration/Advertising
BOB HARRIS

Publisher
DRAGON PUBLICATIONS

Subscriptions
UK £14 for 12 issues
Overseas (surface) £20 for 12 issues

ISSN 0265-177

Address: Dragon Publications, 49 Alexandra Road, Hounslow, Middlesex TW3 4HP, United Kingdom.

Published by Dragon Publications 1988

Typeset by Artset Limited, London NW1

Printed by Headley Brothers Ltd. Ashford, Kent

Registered at the Post Office as a newspaper. Dragon and its logo are trademarks of Eurohard Ltd.

How to submit articles

The quality of the material we can publish in *Dragon User* each month will, to a very great extent depend on the quality of the discoveries that you can make with your Dragon. The Dragon computer was launched on to the market with a powerful version of Basic, but with very poor documentation.

Articles which are submitted to *Dragon User* for publication should not be more than 3000 words long. All submissions should be typed. Please leave wide margins and a double space between each line. Programs should, whenever possible, be computer printed on plain white paper and be accompanied by a tape of the program.

We cannot guarantee to return every submitted article or program, so please keep a copy. If you want to have your program returned you must include a stamped addressed envelope.

Letters

This is your chance to air your views — send your tips, compliments and complaints to Letters Page, *Dragon User*, 49 Alexandra Road, Hounslow, Middx TW3 4HP.

Where is the OS9 group?

THE Dragon disc drives and OS-9 system, which I purchased in March 1988, contains a series of OS-9 user discs bought by the previous owner.

Having read these and found them informative and interesting, I have tried since to join the OS-9 User Group, by telephone and letter, without any success. May I, through your columns, enquire as to how one contacts the group, bearing in mind that I have written and telephoned on numerous occasions the group address? I am particularly interested in acquiring back news discs from number 18 (July 86) onwards as there seemed to be a suggestions that an OS-9 tutorial was going to be produced and this, together with the programs and information contained on the discs, would seem to be excellent value for money as their subscription is only £10.

CAN we have news or up to date information about the OS-9 Users Group, someone, please? I have had two or three etters in recent months (which is two or three more than usual) from people who can't get a response from the Group. Martin Vernon was hard to get hold of in person at the best of times, which is his right and privilege as a hard-working organiser, but I wonder if the address has changed without many of us realising it?

Every month we will be shelling out a game or two, courtesy of our supplies, to the reader/s who send the most interesting or entertaining letters. So send us your hints and your opinions, send us your hi-scores and suggestions. Send us your best Dragon stories. What d'you think we are, mind readers?!

EXTRA PUFF

The Botto revealed

IN the November issue of *Dragon User* you ask where the complimentary reference to the Dragon in an Atari magazine comes from. You will find the quote on page 67 of the July 1988 issue of *Atari ST User*. The article, entitled *A thoroughbred among micros* - referring, alas, to the ST rather than the Dragon - was written by Francis Botto (honestly!), and it was after him that the 'botter' was named. A botter is, of course, an involuntary and inadvertent compliment paid by the supporter of one computer to a competitor's product.

In an obvious attempt to redress the balance, in the October issue of *Atari ST User* the same author disparaged the Dragon in the following words: "Having a complete backup system (to a hard disc) is perhaps beyond most of us, but it's worth keeping a component or two you know that your ST has blown in the past (!), so long as it's nothing more than an inexpensive chip. For example, in the early days, Dragon computers tended to devour sound chips, and BBC micros had a reputation for blowing 74LS245 data buffers."

Roger Wells, 27 Sweda Court
Chesham Street, Brighton
BN2 1NG

Francis Botto ... who does that remind me of? Well, this is a timely reminder that, just because the Atari is larger, newer and more expensive than our Dragon, it doesn't mean that it has better table manners.

Postscript

WITH reference to Brian Cadge's answer to Adrian Orbit on the *Postscript* language, I would also like to add that *Postscript* printers can be connected to parallel ports. At least, mine does!

I have an AST TurboLaser/PS with a Centronics port facility which can definitely be connected on the Dragon. The *Postscript* program indicated by Brian Cadge in *Dragon Answers* is in fact wrong. The **makefont** operator is preceded by the transformation matrix in square brackets, and not the ordinary brackets as given in the listing. Otherwise, the *Postscript* interpreter will not print the desired result. Forthermore, the **'o o moveto' command will drive the string 'Dragon User' ou tof the A4 siz paper. The correct listing of the** *Postscript* program is in fact:

/Helvetica findfont ⌜36 0 0 144 0 0⌝
makefont setfont
14 44 moveto
(DragonUser) show
showpage

If anyone in the Dragon world wants to know more about *Postscript* or laser printers they can get in touch with me.

Sotos Mandalos
2 Kibblewhite Crescent
Twyford
Berks RG10 9AX

PS I am enclosing two printer outputs of my Turbolaser/PS. The first is the above program and the second a fancy printing of the same string.

THANK you, Sotos. For the missing brackets, refer the reply about typesetting translators, somewhere on this page. I wonder if we have them intact this time...

Elite locked

HELP! Help!

Can anybody help with *Elitecalc*. When I use my program and ask it to COPY or REPLICATE a cell it comes p with OPTIONS (V,N,Q) and whatever I enter it locks up. Has anybody else out there got *Elitecalc* and does their program, do the same, or does anybody know what is the cause of the problem?

80B Main Road
Old Duston
Northampton
NN5 6RA

PS Some time ago the OS-9 Group seemed to have disappeared. However, they suddenly reappeared and I as a new member received a disc in March 1988 and then another in June 1988. That is the last I heard from the group. Do you know anything about them?

YES, I remember the first great disappearance. Perhaps Martin intermittently just finds running the group and trying to remain alive and well at the same time a bit too much for him. Refer Paul Grade's column this month (not to mention Paul's regular editorials in Update *for a sidelight on this syndrome. I hope we hear something soon.*

Coco converter

LIKE most CoCo owners I have bemoaned the fact that very little software available for the Dragon will also run on the CoCo, so I have decided to do something about it. I am willing to convert any Basic program and TRY to convert any machine code program, so that they will run on the CoCo.

Now, as most people know, the ROM routines in the Dragon's ROM have different addresses to their equvalents in the CoCo. With this in mind I would be eternally grateful if some kind Dragon owner could disassemble the Dragon's ROM and send me a listing. Of course, I am willing to pay the postage and a reasonable fee for this service.

I have had a few recent successes in conversions, most natably Starship Software's *Composer Companion*, and

Destiny adventures. In addition I feel it only fair to point out that there are some machine code programs I have been unable to convert, for example. programs that add commands to Basic, eg windows, flash, write, R G Whittaker's two extra commands and Peter Whittaker's *GrafText*.

All I ask for this service is the odd acknowledgement, something to cover the cost of printer paper, cassettes, postage etc., and customers to realise that I can only do this in my spare time, so they will have to be patient and not expect a fully working and tested program a week after I have received it. Please enclose an assembly listing with all machine code programs. I regret that I cannot convert disc software.

Paul Marlow
50 Lime Ave.
Bentley
Walsall
W. Midlands
WS2 0JF

Primesearch search party

IT would appear that several errors have crept into my *Primesearch* programs in the November issue of *Dragon User.*

Line 180 should read: 180 A1$=" "

The 330 in line 660 should read 300

Line 710 should read: 710 FLREAD NM$,FROM RR*80,FOR 81,A1$

LINE 740 and 750 should each be incremented by 10 to insert an extra line which reads 740 GOTO 620. I deleted this in error from the program thinking it was superfluous, but it was necessary after all.

I apologise for any brain failure caused by the presence of these errors but hope you managed to sort them out yourself. In passing, may I stress that non-disc drive users should not feel dismayed on seeing the DOS commands in the listings, as they may be easily modified, but at the expense of being able to test a much smaller range of numbers. If any user would like assistance with any aspect of *Primesearch*, then send an SAE to me and I will help you as best I can.

Paul Weedon
Summerleys, Alderley
Wotton-under-Edge
Glos. GL12 7QT

And further:

SINCE I wrote my letter to your letters page about errors in my *Primesearch* listing in November 88, it has come to my notice that there is a further (printers?) error/omission. Hopefully it is not too late to include the following to the list of other errors in my letter. I would be grateful if you would do this for me. Thank you.

Yours faithfully,
Paul Weedon.

(Editor looks into envelope. Nothing falls out. Editor checks letter again. Nothing but an expanse of white paper ...)

NEXT letter:

Dear Helen,

I forgot to write what the ommissions were in my other letter today. Sorry i forgot it in my haste.

The omission was:
590 r$(i)=x$
My apologies again
Yours sincerely
david weedon (son)

(*Thank you for sorting out the old man, David. As this is all written in beautiful Gothic script, it is a little difficult to be sure of the exact correction, as the Goths apparently did not have a symbol for = or $. Let us check the original ... esc save whirr whirr clear click chug chug whirr read bleep! silence ..too late, it's already been wiped. I don't normallly wipe things till I've checked them, but that doesn't mean that the typesetters' translation program won't lose something without me noticing. It does happen, see Harvey Gray's letter elsewhere on this page. On the other hand, I can't see anything in "590 r$(i)=x$" which the machine would falter over, and there are one or two points which bear distressing evidence of having been, er, shall we say, misprints. Send me nice narrow listings with no smudges, Paul, and I won't have to retype'em. Deal?*)

I ink this is alright

ITS a bit late to be answering the query, but yes, I have used the reinking service of Alladink of Eyemouth (see *Dragon User* September 1988. THe response was good, the price reasonable. Of two ribbons reinked so far, one has come back as good as new, the other smudged a little initially, but settled down after a few days. I would have no qualms about recommending them.

Malcolm Cowen
21/23 Bristol Road
Levelnshulme
Manchester 19
M19 3NU

No indeed, your letter comes timely after R A Davies' tale of disaster in his review. The success of this operation depends to some extent on the type of ribbon, and using a worn ribbon can cause problems which are only made worse by reinking. I wrote off a manual typewriter like that, once. Be warned, not worn. By the way, you don't know what the OS-9 User Group is up to these days, do you?

For Pete's sake, Paul

NO, I haven't forgotten to include the disc, I shall be sending the next lot of copy to you in plenty of time for your January issue deadline (*Actually, it arrived on the final copy date, but as Pete sends his stuff in on a disc which I can digest directly, and as it's two days before his birthday, and as I'm running late anyway, no radishes for Pete.*) but having just received and read bits of the October issue the brain was diverted away from penguins and other palatable puzzles just long enough to write this letter.

What prompted finger to keyboard was Paul Grade's column. The phrase "*Atari, Amstrad, Commodore etc. are still selling mediocre products to willing mugs*" was the one that did it. I regularly use computers manufactured by all three, none of which I would regard as being mediocre products. Nor do I think that I'm a willing mug for parting with my money.

I use an Amstrad PCW primarily for word processing, and it helps some magazines (including this one) considerably that I choose to do so. I bought it for just that purpose, and at less than £400 all-in-one, it represents a great bargain. Apart from magazines, I can easily maintain a healthy correspondence with friends up and down the country, which I would be less inclined to do it I didn't have a PCW.

A Commodore 64 lurks nearby, and although that tends to be a 'games' machine, I also do quite a lot of programming on it. It is now entering its seventh year, and is still a great computer for its price.

An Atari 1040St belongs to a friend of mine, and that gets use by me about once a week for shared development work on games. Again, for the price, it is a marvellous computer, it most certainly is NOT a mediocre product, and I wouldn't like to be in Paul Grade's shoes were he to call my friend a willing mug within earshot. Felled by a single glance, most likely.

Then there is the Dragon, which I use for playing adventures. All four computers, in their own way, have a job to do and do it well. Total cost to me, in real terms, is around a thousand pounds spread over two years of ownership in sunny Wigan. In other words, about the price of a packet of twenty cigarettes a day, or two pints of bitter at our northern prices.

So please don't turn *Dragon User* into a forum for one of those meaningless debates on which computer is best. The best computer is the one that you are making good use of at the time, and let that be the end of it.

PS IT would be a great help to me if you could get any pre-prerelease adventures, and I could incorporate everything into one big column, creating a little adventure fanzine within the confines of *Dragon User.*

Pete Gerrard
Sunny Wigan

WELL, I think Pete and friend would be exempted from mugdom entirely on the grounds that they are putting their computers to constructive use. The mug is, after all, the one who is staring at a pile of hardware thinking, why won't this do what I want? I agree about the PCW. If I had more room on my desk I'd have one here, and just use it for bashing out instant letters at the moment I think of them. No more, why don't you write us, Helen?

News desk

If you have any new products for the Dragon — software or hardware — please write to The Newsdesk at 49 Alexandra Road, Hounslow, Middx TW3 4HP.

Dragonfire still stoking

LATEST price list updates from Dragonfire Services: new programs *Hack-it*, tape £3, *Space Trek 1,2 3, Dream Machine, Time Machine Search*, and *The Immortal Strain*, tape and disk, £4 each, *Dragstone and Kids Pack*, now on disc £2, *Sunken Ghost, Diamond Manor* now on disc, £3 each. All prices plus 50p per item p&p (UK), £1.25 per item overseas.

New: *Intelligent Disc Copier*. A simple but ingenious utility that backs up the entire disc (like using a backup command, including the 'hidden' boot routines, etc.) in the minimum number of disc swaps. Designed for single drive users, but will save time on multi-drive copies too. Free copies of Bootstrap (auto boot) and Hi-Tex (improved screen display) on the disc as well. £5.

Dragonfire will arrange a post office special delivery service over Christmas for an extra £1.75 on the price of a whole order. Regrettably this column is unlikely to reach you before Christmas, but no doubt if it is a success Dragonfire will consider continuing the service.

Dragonfire Services, 13 Parry Jones Close, Blaina, Gwent NP3 3NH.

Ethnologist finds new DOS bag

PHILIP Scott has written to us with a bug report on a fault in DragonDOS 1.0, SuperDOS and versions to DOSplus to 4.6. "I have not seen a previous report of this fault, and Paul Grade has not heard of it." writes Philip. "Nor have I", says Bob Harris. Philip blames the bug for corrupting calculations and disc directory tracks.

"It has been found recently that programs performing intensive disc output can corrupt normal calcultions or cause SREAD and SWRITE to access the wrong disc sector. These errors can be both transient (for example, corrupting output from PRINT) and permanent and are difficult to effectively demonstrate or detect.

It is perhaps worth noting that SREAD and SWRITE usually access track 20 (the directory track) when the problem occurs. It is possible to overcome the problem by adding program statements, but each program generally needs a different approach, and the effectiveness of the process can very from run to run.

DOSplus owners who have not been contacted should request exchange details for replacement by DOSplus 4.7."

Philip G. Scott, 4 Badgerwood Drive, Frimley, Camberley Surrey GU16 5UF.

Can anybody shed any light on this?

Radiation maiden



THE young lady in the pinny is wearing an overall designed to protect pregnant women from the possibility of miscarriage due to radiation from VDUs.

Although experts still disagree on whether RF radiation from VDUs (or televisions) can disrupt the growth of embryos in the first weeks of pregnancy, research figures indicate that women working more than 20 hours a week at VDUs are more prone to miscarriage than those doing non-VDU work. A garment like this blocks up to 99% of radiations from most VDUs.

The Microshield overall, shown here, costs £55 plus VAT and is available from computer dealers or from Mediatrade marketing, PO Box 15, Emsworth, Hants PO10 7YH.

Visitext plus, plus

Visitext Plus by Orange Software has been upgraded with better facilities for adding and deleting text, and block moves. Copy can also be laid out in more than one column per page, in magazine style, with the wordwrap and justification working within the columns. Orange Software are expected to offer an upgrade service to existing users.

Orange Software, The Garth, Star Road, Nant-y-Derry, Abergavenny, Gwent NP7 9DP.

From Ken G. Smith

Repair man back

Nic Spiers has announced to Dragon User that, "finally, after 13 months" he has succeeded in moving house, and is now back in business selling spare parts for the Dragon, and undertaking repairs.

Nic's classified ad. (page 27) mentions "transformers, SAMs, MPUs and many other items, repairs and upgrades." For a price/stock list, send an SAE to Nic Spiers at 20 Eaton Way, Great Totham, Essex CM9 8EE.

Update to date

OCTOBER's issue of *Dragon Update* features a page on *Sprite Magic*, a feature on Logo, two or three short programs, much correspondence arguing for/against the Dragon and the Atari ST, a review of the PCW Show (scenes of Bob Preston doing business amid the throng), several short games reviews by Mike Stott, a user note on DosPlus 4 from Philip Scott, and the usual couple of pages from Paul, among other things exhorting us all to suport the shows so that NDUG can go ahead and run one next year. This is Update's 50th issue, but wasn't quite the celebratory edition Paul had hoped for, due to that old favourite the postal strike - cancelled Crosswords and so forth. And the Editor would like some more articles. National Dragon Users Group, c/o Paul Grade, 6 Navarino Road, Worthing, Sussex.

No connection

SIMON Jones would like it known that his New Era Publications and Software company has no connection either with Harry Whitehouse's erstwhile New Era Interface, nor with New Era, publishers of science fiction books.

Lee conquers USA

GORDON Lee has had a good response from readers of Scientific American to his Primegrid puzzle, and we hope to publish an upgraded puzzle resulting from that correspondence in next month's issue.

LAST June I typed in the names and addresses of nearly 2400 subscribers to Dragon User. This issue will go out to just 1450 subscribers.

In recent months, the flow of income from subscriptions has dwindled to a trickle. In particular, since the end of the postal strike, the take-up rate of renewal invoices has fallen from 60-65% to a mere 20%.

As a result, the money in the bank account is draining like bathwater. It has now reached the point where we will not be able to pay for a further issue.

In these circumstances, there is no possibility - and indeed very little point - in attempting to carry on. Therefore this issue of Dragon User will have to be the last.

Whether Dragon Publications will be able to continue in any form is at present unclear. Should any subscription income be left after creditors' claims have been met, I propose to dontate it to the National Dragon Users Group so that the end of Dragon User does not further endanger the survival of the Dragon itself. I would strongly urge all readers who have not already done so to subscribe to NDUG by writing to Paul Grade, 6 Navarino Road, Worthing, Sussex.

I'm sure that readers will wish to join me in expressing thanks to Helen Armstrong and the other contributors for their work over the last few months. I can only add my apologies to those who did renew, and who believed, as I did, that the Dragon community was still capable of supporting a professionally produced magazine. I am only sorry that this has not proved to be the case.

Yours sincerely,

Bob Harris

I am intending to pass on unpublished contributions to Dragon Update - anybody who has sent me an article and doesn't want it sent on, please drop us a line.

Helen

Helen

New software for review should be sent to *Dragon User*, 49 Alexandra Road, Hounslow, Middlesex TW3 4HP

Second-saver for a fine tuned Dragon system

Program: *Express/OS-9.*
Equipment needed: Dragon 64 with the DragonPlus board installed.
Supplier: Compusense Ltd., Willoughby Lane, London N17 0SP. (See also last month's ad.)
Price: £16.95.

A fair number of Dragon 64 owners have upgraded using the Compusense DragonPlus board, the nearest thing to a 128K Dragon which exists. I suspect this was mainly to get the very clear 80 column display, which is a great improvement when using serious software. Another bonus is the extra 64K of memory, which cannot, however, be accessed directly from Basic. A recent issue of DU reviewed the *Edit+* program, and Compusense produced a special version of this which uses the 80-column display and can access the extra memory using a new POKE command. Apart from problems which appeared with the INKEY$ command using this program, POKEing is hardly an ideal way to use an extra 64K of memory.

A much better solution was to use the 64K as a ramdisc with either Flex or OS-9. The extra 64K of memory is first formatted just like a hard disc, and then the user copies the most often used commands from the system disc to the ramdisc. Remember that Flex and OS-9 use almost the full 64K of the Dragon 64 for the programs they run, and any operating commands such as format, copy and DIR are normally picked up from the disc in drive 0 as required. This saves a lot of memory, since the normal Dragon Basic together with DragonDOS takes about 18K permanently away from the maximum of 64K which can be addressed by the 6809 chip. However, it does slow down the operation considerably, and uses the disc drives a lot more. When I first changed from my bashed-up tape recorder to sleek new Cumana discs I marvelled at the almost instantaneous loading of programs and files. Now I resent the few seconds it takes for OS-9 commands to load. Man (and Woman, madame Editor) is never satisfied - this, I suppose, leads to Progress.

The ramdisk drivers available from Compusense and others enabled the commands to be on hand almost instantaneously, providing you had first copied them onto the ramdisk. A special startup file can enable this to be done automatically when BOOTing up, but this prolongs the scratching around on the system disc, during BOOT for at least a further minute, which can be annoying if you only want to use one of the commands. Also, Sod's Law takes a hand - the command you want to use is the only one not copied to ramdisk!

More clever

The new *Express/OS-9* is a much cleverer answer. It uses an advanced cache buffering technique 'only used on large and expensive micros', to quote Compusense. The idea is to keep automatically any command or program you use in a buffer which uses the extra DragonPlus memory, so if you use it again later it will load instantly. Thus, the first time you load the word processor *Stylo* it will take the normal loading time (a little less, actually, for reasons I will divulge) but if you return to Stylo later it loads from the buffer in less than half the time. This assumes you have not used so many other commands in between that you have exceeded the buffer memory of 64K. Not only is there a time saving, but wear and tear on the discs is much reduced. Another clever feature is an option which enables you to store the disc directory track in the buffer. Since this is always accessed before any disc read or write, and often several times during the transfer of large files, much toing and froing of the disc heads can be avoided, which is why even on first load the access is quicker.

Now, is it worth buying a utility to save a few seconds, and are there any snags? Compusense claim that using the C compiler for a short program they took 166 seconds as compared to the normal 272 seconds, and the C compiler is too large for it all to fit in the cache at once, so this is not a 'best case'. If you are frequently using compilers this saving certainly seems worthwhile. The three main snags seem to be that a little of your memory is permanently used by a module called XPR, that you have to initialise by both loading XPR and by calling a longer command called XP, and that the idea of holding the directory in memory can be dangerous to your disc's health! Imagine what would happen if you changed discs and then tried to do a write on the new one using the filemap of the old one! (*I don't have to, David. My Apricot will do something fairly horrid, no problem, if you change discs without clearing the memory first. The Amstrad PCW (Locoscript) won't let you do that - it performs a Disc Change automatically. But it's a slow little chap. Ed.*).

Compusense have addressed the last problem by offering a number of safeguards. Using the command XPR when changing discs clears the buffer (but what if you forget?). The command XP -W/d0 will stop any writes occurring to the specified drive /d0 so the worst that can happen is that you get some funny reads after changing discs. This command is recommended for the drive which contains your system disc if you have at least two drives. However, you will probably want to save quite frequently on your data disc, so this is not an option for the other drive.

Logic check

The command XP C tells *Express/OS-9* to check logical sector zero of the disc each time. This tells OS-9 if the disc has been changed, and if this is detected the directory cache will be rewritten. However, this means the drive always switches on at each command, and some of the timesaving is lost. Also, to guard against two discs having the same name, it is recommended that the supplied utility CLIP is used to timestamp each of your discs using the system clock. CLIP uses the last six bytes on sector zero which are otherwise unused by OS-9. This means each disc (even a backup) is unique. Sector zero checking is the default, which is switched off using the command XP -C.

The XP command has other alternatives. The parameter I initialises *Express/OS-9* while X cancels it. -A/dx deactivates drive /dx while A/dx reactivates it. S displays the status of each drive and D displays the performance so far. This is an interesting one, as it tells you how many disc accesses have been made and what percentage has been saved by *Express*. The counters are then reset, but a similar command P does not reset them. There is a useful HELP command obtained by XP ? or XP H. Both give a list of parameters and their uses.

The disc supplied contains four files, XPR, XR, CLIP and a sample STARTUP file. XP must be copied into your commands directory and you can do the same with XPR, loading it before calling XP, or include it in your boot file using OS9GEN. The startup file does the usual things but also gets you into 80 columns, loads XPR and then calls XP with a number of parameters. These initialise *Express* without sector checking and enable write protect on drive 0, make drive 1 a read/write drive with sector zero checking and disable *Express* on drives 2 and 3. Finally the drive status is displayed. A 12 page A4 instruction leaflet gives clear information apart from a few obvious mistakes.

Since *Express* is not compatible with use of the ramdisc, there is a warning message on loading XPR if the ramdisc drivers are present in memory. This only works for the Compusense ones - if you have the /r0 driver (which has the ad-

vantage of working with the same FORMAT command as discs) no warning is given. *Express* also expects to find devices /d2 and /d3 in memory and gives an error if they are not found. This doesn't seem to affect its use, however, apart from the status command 's' which gives rubbish on the lines which would report on drives 2 and 3.

So much for the theory, what about *Express* in practice? My review copy refused to load the XP module, giving a CRC error. This would, of course, have been due to nasty things which the GPO often delight in doing to our packages, and I only mention it to highlight the fact that a phone call to Compusense resulted in a replacement by return of post. This worked perfectly, doing everything the instruction leaflet said it would, with one exception: the write protect option does indeed give a 'write protect' error number 242 when you try to delete a file on a protected drive. However, when you try to use the COPY command to write on that drive you get an error 215 'syntax error in pathlist'. Not a serious problem, but it could cause some head scratching! I then set out to see what tim^ advantages were to be gained using my new Mitsubishi 80 track double sided drives.

Timings using Mitsubishi 80-track drives

Job undertaken	Without XPR	With XPR using -C option	
		First time	subsequent
Load Stylograph	12s	10s	6s
Load Dynacalc	16s	11s	6s
Compile short C program (copyg)	6 min 0s	4 min 12s	
Compile longer C program (grep)	10 min 0s	7 min 42s	
Compile short Pascal program (del)	3 min 36s	1 min 40s	

Load times for *Stylograph* and *Dynacalc* are increased by 1 sec if -C option is not used.

Compusense say that their Dragon Data drives took 17 secs to load *Stylograph*, and that this was reduced to 12 secs for the first load and 5 seconds for subsequent loadings. Well, we won't argue over one second for the final loading. It is evident that the original Dragon drives were slower than mine, so that the potential benefits are greater. Even so, my loading times were halved and there was a very substantial time saving while compiling both Pascal and C programs. If you usually make a coffee while waiting for the compilation, then this won't grab you as very important, but if (like me) you tend to write programs with errors in them and have to recompile several times to find them, then the time saving is useful.

A very noticeable effect was the reduced head movement of the drives while loading programs and indeed a frequent loading of commands without the drive light coming on at all. This is much quieter and must lead, as claimed, to less wear on the drives and discs. If, like me, you have 80 track or double sided drives, or both, then you can have all your commands and programs on one system disc. In this case use the -W and -C commands on drive 0, and don't remove this disc. It is probably safer not to use the -C option on drive 1 as you will be reading and writing data so you cannot use the -W to safeguard your disc. (*I am against using the same disc for system and work files, for numerous reasons, each one small in itself but adding up to lost files and other inconveniences over a period of time. Ed.*) If I had only 40 track single drives I don't think I would use the -C option at all, since the danger of forgetting to do an XPR when changing discs far outweighs the one second time advantage.

Is it worth the £17 price tag? If, like me, you got your OS-9 stuff at the sell-off prices from John Penn then paying almost as much for a speed-up command as for a C compiler seems expensive, but if you consider the original Dragon Data prices for OS-9, it seems cheap. And remember that Dragon Data prices were much cheaper than comparable software for other computers ... If you do a lot of compiling, or use the system commands a lot, then it has got to be worth buying. If you normally boot up straight into *Stylograph* or *Dynacalc*, and stay in them until you save your final file, then the extra time and memory taken in loading *Express* is probably not worthwhile. In any case, Compusense are to be congratulated on an imaginative and welcome addition to the OS-9 armoury, and so deserve five dragons for what must be the most useful software addition for some time. It also makes the DragonPlus board an even more attractive add-on for the Dragon.

David Rothery



New games new faces

Title: *Mandragore*
Supplier: Kouga Software, 94 The Oval, Firth Park, Sheffield
Price: £4.00

MANDRAGORE is a new arcade type game from a new software house, which just about symbolises the metamorphosis of the Dragon market recently. Gone are the old stalwarts like Microdeal and Quickbeam to be replaced by new producers such as Dragonfire, Orange and in this case Kouga.

With 'gore' tucked away in the title it would appear that this would be a shoot everything game. Not quite the case, as you play a more tactical role. 'You' in this instance are Mandragore, an exploration robot sniffing about in an Egyptian pyramid. Unfortunately for you, a complex defence system is set off, with all kinds of creatures and objects lurking and loitering with intent.

To complete the game you have to conquer two levels each composed of five floors, a floor being like a level in *Kung-fu*, where you progress either in an Easterly or Westerly direction while the screen scrolls in the appropriate fashion.

As for the game's objective, that seems to be to avoid hazards or to shoot them and not only that but to get to the end of level 2 with one your precious five lives intact! I don't know what happens then, as it's a difficult game. The first floor is relatively easy; you move up and down the pyramid lanes shooting mutant aardvarks portrayed in excellent graphics and which transform into equally impressive skeletons once shot. Although once dead they can no longer spit venom at you, to touch them is still deadly, unfortunately. Other objects on the first floor include a fairly tame mummy which fires the odd bullet and small pyramids on the floor which block your way.

In fact there is no aspect of the graphics I can criticise. Once the first floor is completed other well drawn objects and creatures appear, such as swooping eagles and ants which fire relentlessly at you. All this is portrayed in hi-res green-black, which I always think is less visually effective than the Dragon's only other real option, black-buff, but is perhaps less strain on the eyes.

Joysticks are in action again here, with traditional Dragon sticks being more effective than my Atari favourite, as small taps on the stick are needed for up-down movement rather than big lunges.

With *Mandragore* and *Lucifer's Kingdom* being the last two new Dragon programs I've looked at I can safely say that quality software isn't drying up. *Mandragore* is one of the best graphic games I have seen for the Dragon; it's quite fun to lose a life to see yourself fall into a pile of ashes while your cap spins round in the air before falling to earth. The only problem is that I don't find it the most addictive game I've ever played - not quite as addictive as say *Lucifers Kingdom,* but any program so well written can only deserve the accolade of five Dragons.

Philip Stott

Wordsearch

Please get your answers in to *Dragon User Wordsearch Department* by the end of the month on the front cover

Find the world currencies hidden in the grid.

WELCOME to the first Dragon Wordsearch! Now that the Thirteenth Crossword is soon to retire, it has found a worthy successor. All you have to do is find the world currencies - listed on the left - hidden in the Wordsearch grid. The letters remaining when all have been located will spell out, when re-arranged, the name of a Dragon software title. Clue: END OF ENGAGEMENT? (6,2,3,4).

And what of the Twelfth Dragon Crossword, still working hard for a living? It reports back to us, groaning under the weight of its followers.

"We have here Richard Crofts of Willington, who would like anything in 3D or indeed almost anything at all, and A. Carlson of RAF Wittering, who fancies a round of Backgammon.

The phrase is DISC INTERFACE.

Send your answers to the First Dragon Wordsearch to Dragon User. The first correct answers out of the Editor's hat win Something from the Magic Bottomless Box. Try telling us what you'd like - you might be lucky.

T	N	E	C	N	A	R	F	D	I	N	A	R
N	G	K	G	E	U	P	C	T	D	M	T	R
I	R	E	R	P	N	E	E	R	N	A	E	N
R	O	S	E	A	N	T	A	S	D	H	S	E
O	A	E	K	T	M	C	I	N	O	R	E	Y
F	T	G	A	I	H	H	A	M	A	I	P	R
U	P	V	I	M	L	R	C	V	E	D	O	Z
K	O	I	A	N	R	L	I	S	O	P	L	E
C	R	F	A	E	N	L	I	L	T	O	E	S
E	T	O	G	S	O	E	L	N	T	U	K	C
P	L	U	N	B	T	A	F	Y	G	N	E	U
O	R	I	E	A	R	R	K	P	I	D	H	D
K	R	A	M	L	H	N	E	F	N	E	S	O

BOLIVAR
CENT
CENTAVO
CENTIME
DEUTSCHMARK
DINAR
DIRHAM
DOLLAR
DRACHMA
ESCUDO
FEN
FORINT
FRANC
GROAT
KOPECK
KRONA
KRUGERRAND
LEU
MARK
MIL
PESETA
PESO
PFENNIG
PIASTRE
POUND
RUPEE
SHEKEL
SKILLING
YEN
ZLOTY

Screening the Dragon

*Radio amateur **N J Cleaver** lowers the noise threshold*

The following information may be of some use to users of the Dragon, particularly radio amateurs, of which I am one.

I use the Dragon with a single disc drive and monitor together with a Yaesu FT-77 HF transceiver to transmit and receive radio teleprinter (RTTY) software supplied by Grosvenor Software.

As a lot of readers will probably know, the Dragon does tend to generate a rather a lot of radio interference which can cause havoc when trying to receive weak RTTY. In my case this was no exception.

I have tried various methods to eliminate this interference, for example, the use of screened leads to every piece of equipment, and the physical repositioning of computer/monitor and radio. This had some marginal success but I was still plagued by the interference directly radiated from the Dragon.

Having read articles about screening the inside of the computer, I decided that I would set about spraying the inside of the computer case with a conductive nickel screening spray.

The Dragon case comes apart very easily by the removal of the four screws located underneath the case. Next the printed circuit boards are released by the removal of the screws at each corner of the boards. There are two ribbon cables, one connecting the keyboard to the main circuit board and one connecting the power supply/video modulator board to the main circuit boards. Once the screws are removed all three boards can be lifted out of the case.

Next, the inside of the upper and lower parts of the case must be thoroughly cleaned. I used washing up liquid and in no time the inside of both parts was spotless. After a short period of time to allow the case to dry, all the cutouts in the lower half of the case were covered over with tape to prevent overspray. A template was cut to the shape of the keyboard cutout and fixed into place. The final thing to do before spraying is to solder a small length of wire to a small square of Veroboard and to stick the Veroboard to the inside of the case above the position of the power supply board. This will enable earthing of the inside of the case after the spraying has been completed.

The actual spraying part is the easiest and only took a couple of minutes. To prevent overspray I used a large piece of flat card held against the edges of the case. The Veroboard (but not the wire) must be sprayed over too. The nickel spray dries in about ten to fifteen minutes.

When reconstructing the Dragon you must ensure that only 0V (ie earth) parts of the circuit (if any) come into contact with the case. There is only one part of the Dragon keyboard which actually touches the case, that being the left hand side support. This support can easily be scraped clean of nickel spray to prevent any short circuits.

To earth the inside of the case I connected the wire that was soldered to the Veroboard to an earth point on the power supply. This effectively earths the inside of the case. I cannot stress how important it is to check that there is no part of the wiring or the circuit board touching the case. Having connected the wire, the case can be reassembled. The upper half fits snugly into the lower half, making a near perfect earth shield around the circuitry. The shielding cannot be made 100% perfect, as there are cutouts for joystick/printer/cassette ports etc.

When reconnected to my transceiver, the results were very pleasing. A very large reduction in computer generated noise was immediately noticeable. Now many more weak stations can be resolved than before. The problem of TV generated noise from the timebase generator still exists, but it is only a minor irritation compared to the noisy Dragon.

I must state that I and *Dragon User* cannot be held responsible for any accidents or disasters that may befall anyone attempting this modification. The nickel screening spray can be obtained from Maplin Electronics, part no. YM86T, page 486 in the current catalogue.

I would like to add in closing that I very much appreciate *Dragon User*, having every single one since publication started, and I usually find something in every issue that is of interest.

Introduction to Dynacalc

***J B Slinger** introduces the spreadsheet package Dynacalc.*

THERE was a request in the July issue of *Dragon User* for someone to write about *Dynacalc*. *Dynacalc* is a spreadsheet package. A spreadsheet is a big table with lots of 'boxes', called cells, arranged in rows and columns. A spreadsheet package is a computerised version which has been partially pre-programmed. In use, a user completes the programming to suit the particular need of the moment. Putting it another way, a spreadsheet package is an ultra-high level language for programming tabular calculations. I could have said that at the outset but didn't for the reason that spreadsheet packages are always described to potential purchasers in such a way as to minimise the programming aspect. Quite senior managers write or build computer spreadsheets, which they would not do if they thought of the activity as programming! (*Yes, it's true. At first business executives wouldn't use computers because they thought it was typing. True!*) In truth, the language is so high level that the programming is minimal as will be seen.

Locked in cells

The programming consists of typing items into the various cells. The items may be text, or figures, or expressions (formulae). The first two are self explanatory; expressions are given in terms of cell references rather than variables but, this aside, will be understood by any Basic programmer. Thus a cell which contains '+B3*B4' will contain the product of the contents of the two cells known as B3 and B4. Note that if the contents of either of these cells is changed, the computer spreadsheet will instantly recalculate to show the new product. In addition there are a great number of preprogrammed functions; for instance '@SQRT(...)' will evaluate the square root of the expression in the brackets. Traditionally, cell references are given with the convention that a capital letter is used to denote the column and a figure is used to denote a row: cell B3 would be the second cell from the left, and 3 rows down.

At this point is is worthwhile to introduce the word *worksheet* to describe a computer spreadsheet; this is common jargon and obviates any possibility of confusion between a paper spreadsheet and a computer one.

As a tiny example of a worksheet, look at **listing two**. The meaning of each cell will be obvious although the overall objective may be obscure. So before describing the programming, I shall digress to explain the objective, which is to solve Gordon Lee's September puzzle.

Briefly, the September puzzle was to find a vulgar fraction which is a close approximation to the fourth power of pi, the circular constant. From this vulgar fraction we had to produce an approximate value for pi which I will call 'Indian pi' in deference to the man who discovered this approximation. Indian pi had to be accurae enough to give an error of less than 1 inch in the circumference of the earth when compared with the value calculated from the accepted value of pi. I chose to set up my worksheet to follow this statement of the problem. For further exemplification of the worksheet's method I give a Basic program of the same method; see **listing three**. The degree of precision of the Dragon's Basic is barely adequate for the problem but the method should be clear enough. I calculate the diameter of the earth in inches, and thence the circumference of the earth, in lines 30 and 40. I also calculate the 4th power of pi. This has be be converted to a vulgar fraction by trial and error. I multiplied the value of the 4th power of pi by various trial integers (trial denominators) and rounded-off the products to trial integer numerators (see lines 70 and 80). Each numerator/denominator pair was then converted to a trial value of Indian pi and the trial circumference was calculated and compared with the true value. I used the same method in the worksheet except that the trial denominators were put into cell B1 individually. Because *Dynacalc* is accurate to 16 significant digits, the procedure gave an acceptable value for Indian pi and the vulgar fraction.

Now to return to the mechanism for programming the worksheet. I use Flex, so I have to boot the operating system first and then load *Dynacalc*. Actually I do both at the same time since my *Dynacalc* disc has the operating system on it as well and I wrote a startup file to load the package automatically. When loaded, *Dynacalc* displays an empty worksheet with the default column width of 9 characters. This is OK for most purposes, but I had decided that 20 characters width would be more appropriate for my present purpose so I had to change it. This was done by typing '/AWW20Enter'. The slash key / signifies that I want to type a command. The A means that I want to change an 'attribute' of the worksheet, the first W signifies that I want to alter a window (*Dynacalc* has windows!), and the second W that I want to alter the width. There is a question then asking me how many characters wide and the figure 20 is the answer. Entering information into the worksheet is simply by pointing to a cell with the cursor keys, typing the entry and pressing Enter. If the contents of the entry cell are used in an expression in another cell, the worksheet will instantly recalculate itself to reflect the change. Magic!

Well not quite magic because there are three subtleties. The first one is that certain formulae only need to be evaluated once. For example in **listings one and two**, the diameter, the true circumference, and the 4th power of pi are really calculated constants. I chose to leave these as formulae for the purposes of this article, but in a real situation, to save memory and for speed of recalculation, I would want to evaluate them once and for all time. This can be done by entering the expression for the circumference in the form '+72*1760*3960!'.

Exclaim and vanish

The exclamation mark causes the formula to be evaluated (and to disappear) and then pressing the Enter key stores the result in the cell. The second subtlety is that recalculation of a worksheet follows a definite order; it is by column or by row (selected by /ACO for column-wise or /AOR for row-wise recalculation). Whichever is selected, the worksheet recalculates from the top left (cell A1). If any cell has a reference to a cell which has not yet been recalculated the cell will not evaluate correctly. There are situations where forward references have to be used, such as where one is doing an iterative calculation, but one should try to avoid forward references if at all possible. If there are forward references, one can cause recalculation by pressing the exclamation key several times until the values stabilise. The third subtlety is my own, **Listing one** is the natural or default format in that it shows the figures not the formulae. However, I prefer to build a worksheet in the format of **listing two**2. You can switch between the two formats with a toggle command, /WD. There is supposed to be a way to have some cells in formula mode and some in value mode but I have not found it; also there is supposed to be a way to protect certain cells to prevent inadvertent corruption of a worksheet and I have not found that either. Both of these features are desirable as it is as easy to corrupt a worksheet as it is to build it. Pay attention to this if you build a worksheet for someone else to use.

Good presentation

I should now like to review the advantages of using a worksheet. I have already mentioned the 16 digit precision. It is quick to set up a worksheet; the specimen probably took less time than reading this article will take you. It is easy to get a good presentation of a calculation; think how nice it would be to put the data, the method, and the results all in one table for a research report. Irrespective of the calculation feature I know secretaries who use a worksheet simply because it is the quickest way to type a table (trival but time

is money!). There is a rudimentary graphing feature which might be useful. If you have figures in a wide column, /WFP will convert the figures into a horizontal bar chart with a hash sign for every integer value. A specimen is given as **listing four**. There are many built-in functions, see **listing five**; note I have only tested those used in this article. There are sorting and copying facilities too. These are extremely useful in programming a worksheet. I am preparing a further article in which they will be demonstrated. It is not necessary to type similar formulae into dozens of cells. There is a definite art in using worksheets; what to do is usually obvious but there is a substantial amount of 'show-how' involved in doing it with the minimum of fuss. For this reason I am unashamed to advise anyone who is seriously interested by this article to read a good book. There is a strong family resemblance between *Dynacalc* and the most successful spreadsheet package, *Lotus 1-2-3*. There are books about the latter. I have the first edition of *Using 123* published by the Que Corporation. Try to borrow a copy from a library.

I am willing to reply to specific questions if anyone cares to write to me; my address is 17 Taplin Way, Tylers Green, High Wycombe, Bucks. HP10 8DW. Next month (or thereafter) I hope to continue with a simple example of how to construct a worksheet.

```
LISTING 1; FIGURES
          9-8-88     PAGE    1


DENOMINATOR                          22

PI                      3.141592653589793
DIAMETER IN INCHES              501811200
REAL CIRCUM            1576486379.409078

,4TH POWER OF PI         97.4090910340024
ROUND NO?              2143.000002748053
NUMERATOR                            2143
INDIAN PI                3.14159265258265
DIFF OF CIRCUMS          .505395770072937

LISTING 2; FORMULAE
          9-8-88     PAGE    1


DENOMINATOR             22

PI                      3.141592653589793
DIAMETER IN INCHES      72*1760*3960
REAL CIRCUM             B3*B4

,4TH POWER OF PI        B3*B3*B3*B3
ROUND NO?               B1*B7
NUMERATOR               @INT(.5+B8)
INDIAN PI               @SQRT(@SQRT(B9/B1))
DIFF OF CIRCUMS         B5-(B10*B4)

LISTING 4; DYNACALC GRAPH
          9-8-88     PAGE    1


          1 #
          2 ##
          3 ###
          4 ####
          6 ######
          8 ########
         10 ##########
          8 ########
          6 ######
          4 ####
          3 ###
          2 ##
          1 #
```

```
10 PRINT#-2, "LISTING 3; BASIC PROG"
12 PRINT#-2,"CALCULATION OF INDIAN PI"
13 PRINT#-2
14 PRINT#-2,"DENOM NUMER   DIFFERENCE"
20 PI = 3.1415926535
30 DI = 72*1760*3960:'DIAMETER
40 CR = PI*DI:'CIRCUM IN INCHES
50 P4 = PI*PI*PI*PI
60 FOR DE = 15 TO 25:'DENOMIN.
70 X = P4*DE
80 NU = INT(0.5+X)
90 IP = SQR(SQR(NU/DE)):'INDIAN PI
100 DF=CR-(DI*IP): 'DIFFERENCE
105 PRINT #-2, DE;" ";NU;"  ";DF
110 NEXT
```

```
LISTING 3; BASIC PROG
CALCULATION OF INDIAN PI

DENOM NUMER   DIFFERENCE
 15   1461     36783.5
 16   1559    -114932
 17   1656    -10818.5
 18   1753     81745.5
 19   1851    -48394
 20   1948     36783.5
 21   2046    -78813.5
 22   2143    -.5
 23   2240     71971
 24   2338    -30650.5
 25   2435     36783.5
```

LISTING 5

```
1.    Arithmetic type Functions

@ABS( )                Absolute Value, as in Basic
@ACOS( ) @COS()        Cosine, as in Basic
@ASIN( ) @SIN( )       Sine, as in Basic
@ATAN( ) @TAN( )       Tangent, as in Basic
@AVERAGE(list)         The mean value of numeric cells in
                       the list
@COUNT(list)           Number of numeric items in list
@EXP( ) @LN( )         Natural logarithm, as in Basic
@INT( )                Integer value, as in Basic
@LOG( )                Log. to base 10
@MAX(list) @MIN(list) Returns the largest or smallest
                       value from the list
@NPV(list)             Net Present Value of items in list
                       Used in financial work
@PI                    The circular constant
@RND()                 Random number generator
@ROUND(d,n)            Rounds off figure n to a degree
                       specified by d which is a power of
                       10. Note alters the number as
                       opposed to its appearance
@SQRT( )               Square root, as in Basic
@STDDEV(list)          Standard deviation of numeric items
                       in list
@SUM(list)             Sum of numeric items in list

2.  Lookup type functions

@CHOOSE(n,list)        Chooses nth item from list.  "n" is
                       usually the contents of a cell
@INDEX(n,list,z)       Lookup function based on exact
                       matches between n and an item
                .      in the list; returns item offset
                       from the matched item into column
                       or row z
@LOOKUP(n,list,z)      Very similar to @INDEX but based on
                       greater than test

3.    Logic type Functions

@AND(list)             Returns true if all are true
@EOR(2 items)          Returns true if 1 is true
@ERROR                 Forces logical "error",
@IF(test,true,false) Returns contents of true cell or
                       of false cell depending upon
                       whether test cell is true
@ISERROR(cell)         Test if cell has logical "error"
@ISNA(cell)            Test if cell is empty
@NA                    Forces logical "not available"
@OR(list)              True if 1 or more are true
@FALSE                 Returns logical "false"
@NOT(item)             Reverses truth value
@TRUE                  Returns logical "true"

Dynacalc by J B Slinger
```

Basic Monitor

***Craig Henderson** gets inside his Dragon's memory and operates*

THIS Monitor program was written to aid me in examining the contents of my computer's memory and performing various tasks on it. I am very new to Machine Code and I own Alldream from Grosvenor Software. Although this has a monitor I found it was very tedious to load the whole thing in each time I wanted to use only the monitor so I decided to write my own.

I realise that Peter Whittaker published a monitor written in machine code in May 1987, but I find one written in Basic more convenient not only to type in but to leave more memory space for the machine code which is the subject of your interrogation, and it is quicker to LOAD off cassette.

When you run it you will get a title page and then will be asked if you want to load some Machine Code off cassette. After that you will be asked the width of your printer and then go into the main program. If you have not got a printer then just enter 0 to the width.

Operations

The program has eight operations which are called from the main menu by pressing the appropriate key (the ones shown in brackets here). These operations are: (A) Examine memory with autoscroll, (C) Copy a block of memory, (D) Display page (block) of memory, (E) End program, (M) Modify block of memory, (O) Output hex dump, (P) Poke one location, (V) Verify block of memory. (A) Examine memory with auto-scroll. You will be asked to enter the start address as with all the routines, and the screen will fill up line by line. The speed of this can be altered by pressing keys 0-9; 0 is a pause, 9 the fastest and all others in between respectively. When you enter this mode, there will be 2-digit hexadecimal values on the left next to the address and on the right are the corresponding ASCII characters. These characters can be made to disappear and re-appear by the pressing of (W). To quit the routine and return to the main menu press (Q).

(C) Copy of a block of memory. On calling this routine you will be prompted for the start and end addresses of the block to be copied, followed by the start address to be copied to. A limit of 1000 bytes at a time is on this routine which I find plenty. If you selected this routine accidently, enter the start and end addresses as the same value and you will return back to the menu. (D) Display block of memory with only the address of the byte in the top left hand corner of the screen displayed. The address of this block can be changed by pressing the UP and DOWN arrow keys. (E) End program. This re-boots the machine so be sure not to use it until you have a safe copy on tape or disc. (M) Modify Block. You are first asked for the start and end addresses and then if the same value is to be put in all the locations within the block. If (Y) then you enter the value and the computer will do the rest, but if (N) you are asked to enter them all individually one after another and the address is shown. (O) Output dump in hexadecimal with checksum to screen or printer. The start and end addresses are prompted for ,and then if the data is to go to the screen (0) or the printer (2). Then the data is either dumped straight onto the printer or to the screen with auto-scroll and at the end a small pause before returning to the menu. This can be paused anywhere by using the usual <SHIFT> and <@>. An example of this dump is shown in listing 2. (P) Poke a single location with a value. You will be asked for the address and then the value and then you will return to the menu. (V) Verify a block of memory. Following the input of the start and end addresses, the computer will display the current location and value at which it is working. If the location is OK then it will go on to the next location, but if there is something wrong (or that area is ROM) then you will get a failed message with details.

I do hope this is of use and you enjoy using it. If you do not like the idea of typing this in then a copy is available from me on cassete at the price of £2. Send a cheque to Craig Henderson, 'Sanray', 39 Woodbury Avenue, Wells, Somerset, BA5 2XW.

```
0 CLS
1 PRINT"  ****************************    **
                  **    ** A MACHINE CODE MONITOR
**    ** WRITTEN IN BASIC, THUS **    **  LEAVING
MORE  MEMORY  **    **  FOR THE MACHINE CODE  **
  **    TO BE WORKED ON.    **"
2 PRINT"  **                           **    ********
********************    **
**    **  WRITTEN BY CRAIG D.M  **    **          HE
NDERSON       **    **                          **
  **    JANUARY 3/4 1987    **"
3 PRINT"  **                           **    ********
********************";
4 SCREEN 0,1:FOR O=1 TO 2000:NEXT
5 ' *** LOAD M/C ? ***
6 CLS
7 PRINT"DO YOU WISH TO LOAD SOME M/C    BEFORE CON
TINUING WITH THE      PROGRAM ?"
8 I$=INKEY$:IF I$=""THEN 8
9 IF I$="N" THEN 16
10 IF I$<>"Y" THEN 8
11 PRINT:PRINT:INPUT"WHAT IS THE START ADDRESS OF
THEM/C PROGRAM TO BE LOADED ";ST
12 IF ST<>0 THEN CLEAR 200,ST-1
13 PRINT:PRINT:INPUT "WHAT IS THE FILENAME OF THE
M/C TO BE LOADED ";F$
14 CLS:PRINT@231,"SEARCHING/LOADING"
15 CLOADM F$
16 ' *** SET PRINTER WIDTH ***
17 CLS:INPUT"WHAT IS THE WIDTH OF YOUR      PRINT
ER ";L
18 POKE 155,L:POKE 328,0
19 GOTO 22
20 I$=INKEY$:IF I$="" THEN 20
21 RETURN
22 DIM C(1000)
23 POKE 65495,0:W=0
24 ' ** F=255 - FAST **
25 ' ** W=255 - WITHOUT ASCII **
26 GOTO 156
27 PRINT"START ADDRESS = ";:LINE INPUT ST$:ST=VAL(
ST$):ST=INT(ST):RETURN
28 ' **** 'A' PRESSED ****
29 CLS:PRINT"    AUTOMATIC SCROLL LISTING":PRINT
30 GOSUB 27
31 IF F=0 THEN POKE 65494,0 ELSE POKE 65495,0
32 IF W=0 THEN P=6 ELSE P=8
33 ADDR=ST:PRINT
34 H$=HEX$(ADDR)
35 IF LEN(H$)<4 THEN H$="0"+H$:GOTO 35
36 PRINT"$";H$;" :";
37 FOR O=ADDR TO ADDR+P-1:H$=HEX$(PEEK(O))
38 IF LEN(H$)<2 THEN H$="0"+H$
39 PRINT H$;" ";:NEXT
40 IF P=6 GOSUB 46
41 IF P=8 THEN PRINT
42 I$=INKEY$:IF I$<>"" THEN GOSUB 48
43 IF I$="Q" THEN RETURN
44 FOR M=1 TO PAUSE:NEXT
45 ADDR=ADDR+P:GOTO 34
46 PRINT" ";
47 FOR O=ADDR TO ADDR+P-1:IF PEEK(O)>=33 AND PEEK(
O)<=122 THEN PRINT CHR$(PEEK(O));:NEXT:RETURN ELSE
 PRINT".";:NEXT:RETURN
48 IF I$="Q" THEN RETURN
49 IF I$="W" AND P=6 THEN P=8:RETURN
50 IF I$="W" AND P=8 THEN P=6:RETURN
51 IF I$<"0" OR I$>"9" THEN RETURN
52 IF I$="0" GOSUB 63:RETURN
53 IF I$="1" THEN PAUSE=2000
54 IF I$="2" THEN PAUSE=1778
55 IF I$="3" THEN PAUSE=1556
56 IF I$="4" THEN PAUSE=1334
57 IF I$="5" THEN PAUSE=1112
58 IF I$="6" THEN PAUSE=890
59 IF I$="7" THEN PAUSE=668
```

```
60 IF I$="8" THEN PAUSE=335
61 IF I$="9" THEN PAUSE=1
62 RETURN
63 I$=INKEY$:IF I$<>"" THEN 48 ELSE GOTO 63
64 ' *** 'C' PRESSED ***
65 CLS:PRINT"      COPY A BLOCK OF MEMORY":PRINT
66 INPUT"WHAT IS THE FIRST BYTE OF MEMORYTO BE COP
IED ";ST
67 INPUT "WHAT IS THE LAST BYTE OF MEMORY TO BE CO
PIED ";EN
68 IF ST-EN>1000 THEN PRINT"TOO MANY BYTES":FOR O=
1 TO 1000:NEXT:GOTO 66
69 PRINT:PRINT:INPUT"WHAT IS THE FIRST ADDRESS OF
THEBLOCK YOU WOULD LIKE TO COPY TO ";ADDR
70 P=0:FOR O=ST TO EN:P=P+1:C(P)=PEEK(O):NEXT
71 P=0:FOR O=ADDR TO ADDR+(EN-ST):P=P+1:POKE O,C(P
):NEXT
72 RETURN
73 ' **** 'D' PRESSED ****
74 CLS:PRINT"      DISPLAY BLOCK OF MEMORY":PRINT
75 GOSUB 27
76 ADDR=ST
77 H$=HEX$(ADDR)
78 IF LEN(H$)<4 THEN H$="0"+H$:GOTO 78
79 IF ADDR<0 THEN ADDR=0
80 CLS:PRINT "address";CHR$(128);"at";CHR$(128);"t
op";CHR$(128);"left";CHR$(128);"is $";H$
81 FOR O=1 TO 14:FOR I=ADDR TO ADDR+10:H$=HEX$(PEE
K(I))
82 IF LEN(H$)<2 THEN H$="0"+H$
83 PRINT H$;" ";:NEXT I
84 PRINT@O*32+31,RIGHT$(H$,1);:ADDR=ADDR+11:NEXT O
85 I$=INKEY$:IF I$="" THEN 85
86 IF I$="Q" THEN RETURN
87 IF I$="^" THEN ADDR=ADDR-308:GOTO 77
88 IF I$=CHR$(10) THEN 79
89 GOTO 85
90 ' *** 'M' PRESSED ***
91 CLS:PRINT"      MODIFY BLOCK OF MEMORY":PRINT
92 INPUT "WHAT IS THE START ADDRESS OF THEBLOCK YO
U WISH TO MODIFY ";ST
93 PRINT:PRINT
94 INPUT "WHAT IF THE END ADDRESS OF THE  BLOCK YO
U WISH TO MODIFY ";EN
95 IF ST=EN OR ST>EN THEN RETURN
96 PRINT:PRINT
97 INPUT "DO YOU WISH ALL THE VALUES IN   THE MODI
FIED BLOCK TO BE THE    SAME AS EACH OTHER ";Y$
98 IF LEFT$(Y$,1)<>"Y" THEN GOTO 102
99 PRINT:PRINT
100 INPUT "WHAT IS THE VALUE TO FILL THIS  BLOCK O
F MEMORY WITH ";V
101 FOR O=ST TO EN:POKE O,V:NEXT:RETURN
102 FOR O=ST TO EN:PRINT USING"£££££";O;
103 H$=HEX$(O)
104 IF LEN(H$)<4 THEN H$="0"+H$:GOTO 104
105 PRINT"  ";H$,">> ";
106 LINE INPUT V$:V=VAL(V$)
107 POKE O,V:NEXT
108 RETURN
109 ' *** 'O' PRESSED ***
110 CLS:PRINT" OUTPUT WITH CHECKSUM -- IN ALL":PRI
NT
111 PRINT:GOSUB 27
112 PRINT:PRINT:LINE INPUT "END ADDRESS = ";EN$:EN
=VAL(EN$)
113 IF ST>EN OR EN=ST THEN RETURN
114 PRINT:PRINT
115 INPUT "SCREEN (0) OR PRINTER (2) ";N
116 IF N<>0 AND N<>2 THEN 115
117 IF N=0 THEN P=18
118 IF N=2 THEN P=PEEK(155)-15:POKE 65494,0
119 IF N=2 AND P/2<>INT(P/2) THEN P=P-1:GOTO 119 E
LSE P=P/2
120 FOR O=ST TO EN STEP P
121 T=0:H$=HEX$(O)
122 IF LEN(H$)<4 THEN H$="0"+H$:GOTO 122
123 PRINT£-N,"$";H$;" :";
124 FOR PL=0 TO P-1:H$=HEX$(PEEK(PL+O))
125 IF LEN(H$)<2 THEN H$="0"+H$
126 T=T+VAL("&H"+H$):PRINT£-N,H$;:NEXTPL
127 PRINT£-N,"= $";:H$=HEX$(T)
128 IF LEN(H$)<3 THEN H$="0"+H$:GOTO 128
129 PRINT£-N,H$
130 I$=INKEY$:IF I$="" THEN 132
131 IF I$="Q" THEN FOR O=1 TO 1000:NEXT:RETURN
132 NEXT:FOR O=1 TO 1000:NEXT:POKE 65495,0:RETURN
133 ' *** 'P' PRESSED ***
134 CLS:PRINT"  POKE A LOCATION WITH A VALUE":PRIN
T
135 PRINT"SURE ?":I$=INKEY$
136 I$=INKEY$:IF I$="" THEN 136
137 IF I$<>"Y" THEN RETURN
138 PRINT:PRINT:INPUT"WHAT LOCATION IS TO BE POKED
 ";LOC
139 PRINT:INPUT"WITH WHAT VALUE ";V
140 POKE LOC,V:FOR O=1 TO 1000:NEXT:RETURN
141 ' *** 'V' PRESSED ***
142 CLS:PRINT TAB(10);"VERIFY BLOCK":PRINT
143 GOSUB 27
144 PRINT:PRINT"END ADDRESS = ";:LINE INPUT EN$:EN
=VAL(EN$):EN=INT(EN)
145 IF ST>EN OR ST=EN THEN RETURN
146 FOR O=ST TO EN:P=PEEK(O)
147 PL=0
148 PRINT@192,"WORKING @ LOC $";HEX$(O)
149 PRINT@223,"              VAL $";HEX$(PL)
150 POKE O,PL:IF PEEK(O)<>PL THEN PRINT O;" $";HEX
$(O),"  FAILED ON";PL:POKE O,P:GOSUB 20:IF I$="Q"
 THEN RETURN ELSE GOTO 152
151 PL=PL+1:IF PL<256 THEN 148 ELSE POKE O,P
152 NEXT
153 PRINT"TEST COMPLETED. PRESS A KEY"
154 I$=INKEY$:IF I$="" THEN 154 ELSE RETURN
155 ' ** COMMAND DEFINITIONS **
156 CLS
157 PRINT@11,"main";CHR$(128);"menu"
158 PRINT"A ... EXAMINE MEMORY WITH              AU
TO-SCROLL"
159 PRINT"C ... COPY A BLOCK OF MEMORY"
160 PRINT"D ... DISPLAY BLOCK OF MEMORY          WI
TH ONLY THE ADDRESS OF        THE BYTE IN THE TOP
LEFT        OF THE SCREEN"
161 PRINT"E ... END PROGRAM"
162 PRINT"M ... MODIFY BLOCK OF MEMORY"
163 PRINT"O ... OUTPUT DUMP IN HEX WITH          CH
ECKSUM TO SCREEN OR           PRINTER"
164 PRINT"P ... POKE 1 LOCATION WITH A           SP
ECIFIED VALUE"
165 PRINT"V ... VERIFY MEMORY BLOCK";
166 SCREEN 0,1
167 I$=INKEY$:IF I$="" THEN 167
168 IF I$="A" GOSUB 28
169 IF I$="C" GOSUB 64
170 IF I$="D" GOSUB 73
171 IF I$="E" THEN POKE 65494,0:CLS 0:EXEC 46080
172 IF I$="M" GOSUB 90
173 IF I$="O" GOSUB 109
174 IF I$="P" GOSUB 133
175 IF I$="V" GOSUB 141
176 GOTO 156
```

```
$8000 :7EBB407EBB887EBBE57EBBB57EBCAB7EBD1A7EBD527EBDCF7EBDDC= $102E

$801B :7EBE687EBE127EBDE77EBDAD7EBDA57EBE7B7EBE7C7EBE7D464FD2= $F6B

$8036 :47CF5245CDA7454C53C549C6444154C15052494ED44FCE494E5055= $BD9

$8051 :D4454EC44E4558D44449CD524541C44C45D45255CE524553544F52= $B99
```

```
$806C :C55245545552CE53544FD0504F4BC5434F4ED44C4953D4434C4541= $B1F
$8087 :D24E45D74445C6434C4F41C443534156C54F5045CE434C4F53C54C= $B54
$80A2 :4C4953D45345D452455345D4434CD34D4F544FD2534F554EC44155= $B3D
$80BD :4449CF455845C3534B4950C64445CC454449D454524FCE54524F46= $AF7
$80D8 :C64C494EC550434CD3505345D45052455345D45343524545CE5043= $B02
$80F3 :4C4541D2434F4C4FD2434952434CC55041494ED44745D45055D444= $AEE
$810E :5241D750434F50D9504D4F44C5504C41D9444C4F41C452454E55CD= $B0B
$8129 :544142A854CF5355C246CE544845CE4E4FD4535445D04F46C6ABAD= $CAF
$8144 :AAAFDE414EC44FD2BEBDBC5553494EC7844885B9861686168616 86= $D4C
$815F :478613903D8675872B853288298A8B877786BC85A5851485F38539= $C10
$817A :8E9D85608EAA857184159C81B6D4B682B828B64C8EA4B9D2BA03BA= $ECC
$8195 :5FB981BA9ABADFB770B81E9D6199659AD99ADAA749A8C0A6EFA6F3= $10EC
$81B0 :A9FEAA19A8D4B238AC87AAF0AAF3B051AABEA9AFADBDA0499DFA53= $1138
$81CB :47CE494ED44142D3504FD3524EC45351D24C4FC74558D05349CE43= $C9E
$81E6 :4FD35441CE4154CE504545CB4C45CE535452A45641CC4153C34348= $BCE
$8201 :52A4454FC64A4F595354CB4649D8484558A44C454654A452494748= $A9D
$821C :54A44D4944A4504F494ED4494E4B4559A44D45CD5641525054D249= $AAB
```

Letters

This is your chance to air your views — send your tips, compliments and complaints to Letters
49 Alexandra Road, Hounslow, Middlesex TW3 4HP

Continued from page 3

PLEASE find enclosed a copy of the screen dump routine for the Brother HR5, which was incorrectly printed in the July 1988 issue. Hopefully this one will get in without the mistakes. NB. If anybody wants it placed on disc or cassette then I'll gladly do this, provided they send an SAE and disc or cassette.

Thanks for everything and it is only little errors that creep in so keep up the great work.

Harvey Grey
Three Trees
Greens Farm Lane
Billericay
Essex CM11 2NY

RIGHT, well we have the listing ... it's a bit faint and folded, so I'm going to copy it out and print it, instead of putting it through the typesetter. Ten minutes ... Right, we now have a 'listing that corresponds to Harvey's one. And here it is:

```
10 DIM A (8,1)
20 FOR K=0 TO 8=READ A(K,0) ,A(K,1) :NEXT
30 DATA 3,3,2,1,0,0,3,1,3,3,0,0,1,2,3,1,3
,3
40 PMODE4:SCREEN1,0
50 PRINT£-2,CHR$(27) ;"A";CHR$(7)
60 FOR L=0 TO 255 STEP 4
70 PRINT£-2,CHR$ (13) ;CHR$ (27) ;CHR$ (1
28) ;CHR$ (1) ;
80 FOR K=191 TO 0 STEP -1
90 TO=0:S=0:FOR M=0 TO 3:P=PPOINT(L+M,K)
 : T=T*4+A(P,0) :S=S*4+A(P,1) :NEXT
100 PRINT£-2,CHR$(T) ;CHR$(S) ;
110 NEXT K.L
```

Martin AMJ's last word

Regarding Roger Merrick's article in the November issue about the Coco in Britain, I can say that in our Multi Computer Club we have the CoCo3.

Now with four computers in the house, Dragon 32 upgraded to 64, a Dragon 64, a CoCo 2 64 and a CoCo 3 (128K, needs upgrading to 512K but due to high costs of the required chips, still 128K. Under OS9 I try to switch my three dirves: a Mitsubishi 5.25 (40-80), two NECs 3.5 by Cumana frp, tje Dragpm 64 OS-9 level 1V.2 (the Eurohard Jordi Palet's baby), to the CoCo 3 OS-9 level 2V.2.01 by Tandy. And it works, but there are problems.

For instance, the formats and sizes of the floppies/diskettes is not a point for the Dragon, but due to the older machines, the CoCo 3 has still to ride on the old 35 track ... I have tried to put *Stylo* and RMS under Level 2 on this CoCo 3, but no luck so far, maybe you guys know tha answer to that. Jason Shouler, where are you?

It is a pity, to conclude, that the manufacture of the Dragon has stopped. If we had the kind of video chip from a CoCo 3 installed in the Dragon, a sort of raster memory manager, than it would be a Super Colour Computer, because the old beast has still nowadays features which the CoCo 3 lacks.

PS and I am still waiting for this kind of machine, a Dragon 512, which was promised to me by Eurohard's Technical Manager, to have a review of it. So fulfill this one, Eurohard, or am I speaking to a gone spirit ...?

Martin AMJ Van Wamelen,
3 Lynmetestraat,
Oedelem,
Belgium.

Funny you should say that, Martin . . .

New software for review should be sent to *Dragon User*, 49 Alexandra Road, Hounslow, Middlesex TW3 4HP

A choice of words

Program: *Editext*
Supplier: R & A J Preston, Kings Hall Court, St. Brides Major, Mid Glamorgan.
Price: £6.95

PRESTONS are continuing to produce re-runs of good software and breathe new life into them by halving the cost of the package. *Editext* is their latest offering, being originally produced by Nectarine at £12.95. Please do not dismiss this program on the basis that at £6.95 it cannot be any good, quite the contrary! *Editext* is a versatile word processor providing most of the facilities found in very much more expensive word processors, and can be operated via disc or cassette. The disc version is called *Deditext*, and is on the reverse of the tape. The main features allow normal or justified printing, automatic wordwrap, and include full editing facilities. The latter are fairly comprehensive, enabling letter, word or combination of words to be added, deleted or changed. There is even a Find and Replace routine which couples the Move Lines and Paragraphs to anywhere in the text, and makes form design, or standard letters requiring customisation, very easy.

Additionally, the program allows upper and lower case printing to any required typewidth, print formatting to include multiple top copies, and various typefaces.

The standard choice of print fonts is:

a) Elite (12 characters per inch, enabling more text to be printed in a given line width).

b) Double print (prints the line in two passes of the head, minutely advancing the paper, thickening the dots to achieve a higher print quality).

c) Emphasised print (each dot is printed twice to produce the effect of bold type).

d) Enlarged print (gives the impression of double-height characters printed in bold type - very useful for headings and title pages).

e) Condensed print (prints 16.5 characters to the inch, compared with the normal 10 characters to the inch, or 12 using Elite type - very useful for half-sized documents and wide tabulated data).

You can of course mix typefaces if required by printing the document in sections, using the Change Type Face menu incorporated into the program, which will also automatically revert to the normal Pica type.

Once you are satisfield with the layout and content of the actual print fonts you can then execute a draft print which prints out the text line by line as it is stored in the Dragon's memory. Each of the lines is numbered and all carriage returns are marked. When you have completed your draft print or even if you did not require it at all, you can enter the Formatted Print routine which enables you to format the text for printing to virtually any typewidth up to the maximum allowed by your printer, and with any desired number of lines per page. The text can be justified, margins altered, and line spacing adjusted. Of course, if you wish to save the text to tape or disc there is a very useful facility for doing so, which incorporates a special header file, identifying the text, into the filing routine.

The instructions for the operating of *Editext* are clear, concise and simple and also contain a prompt chart to assist you in your early running of the program. As if all this were not enough, there is even a short appendix outlining examples of how to save time using the facilities overall.

Suffice to say this program is very adaptable, easy to use and given all the facilities it contains, at £6.95 it must rate as one of the best buys of the year, and receives my wholehearted endorsement.

R L N Hewson



The ball rolls again

Program: *Rollaball*
Supplier: R & A J Preston, Kings Hall Court, St. Brides Major, Mid Glamorgan.
Price: £? (cassette)

THIS is not a review of the same piece of software that has been reviewed before in *Dragon User* . As yet I do not even know if it will replace the original *Rolaball* or be called *Rolaball2* or whatever.

When I first recieved *Rolaball* for review from Helen many moons ago I was delighted as I had seen the game, before it was finished, at the Ossett Show in 1987. After many attempts at loading it I eventually got it to work only to find that my fingers were not nimble enough to work the keyboard. Having accessed the cheat screen (hands up those of you who don't know it's there) I set the colours to red and green and also stopped the cubes moving around. This made the task of getting around the screen much easier. Then I set about mapping the 49 screens while my youngest son David took over on the keyboard.

Disaster struck in the form of an unmentionable and I was still experiencing loading problems from the cassette. Having contacted the author, Jonathan Cartwright, I was sent a disc version, but came upon the same unmentionable. After several phone calls to Jonathan, he eventually agreed with me that I was right although we both agreed that probably nobody would be able to get that far playing the game normally. Now I have received the updated version and this is what the original should have been.

To those of you who do not know the game at all I will give you a brief desciption. You guide a ball along a 3D landscape collecting 49 jigsaw pieces which form a picture in the top left hand corner of the screen as you pick them up. To move from one screen to another you select your exit, move on to it and press the appropriate key. At the start of the later version you are presented with a menu from which you can define your own keys for the directions, picking up etc. and this is a definite advantage over the original. There is no joystick option in either version of *Rolaball*, and this is because the ball moves over some narrow scenery and it would be even more difficult than using the keyboard. You can also select the colour set and the speed of play also improvements on the original.

The scenery in this version is slightly different to the original and a score of 100 points is now awarded for each jigsaw piece collected, whereas there was no score feature in the original game. At first I kept getting caught by the enemy cube as I arrived on a new screen but if you watch closely you might discover how to avoid this. Several of the jigsaw pieces appear to be impossible to collect at first but a little imagination and experimentation should help you to get them all. One piece in particular is very difficult to see but I can assure you that it is there. When (or rather if) you have collected the 49 pieces and the jigsaw is complete then a flying saucer appears overhead and the ball rises into it.

There is the usual Cartwright musical accompaniment and an unusual method of clearing the screen which I really enjoyed. I must thank my son David once again for doing most of the work. To sum up I found it to be a challenging game which has been greatly improved from the original.

If I had not seen the game so long ago I would probably have thought that Jonathan had taken parts of *Marble Madness* and *Airball* and put them together to make one game. I think that this is now a very good game, albeit very difficult to get very far into, let alone complete, unless you can find the pokes for unlimited lives or for stopping the cubes moving about.

Mike Stott



Quick on the Draw

***Steve Taylor** draws the line — from assembler, fast.*

THIS routine is designed to replace the Dragon's LINE command for use from assembly language in PMODE 4. I was prompted into writing it by Brian Cadge's reluctance to explain how to use the rom routine from assembler. Although it can be done, it's so complicated and messy that I agree with him entirely — it's not worth the effort.

The rom routine works in any PMODE, and in any valid colour too, but this program is configured for PMODE 4 only, although it does work for colour 0 on 1 or colour 1 on 0 using either SCREEN 1,0 or 1,1. The reason for this restriction is that anyone attempting to write graphics programs in assembler has only one motive — speed. Generalising the program for multi-colour and several modes would slow it down too much for it to be of any advantage.

Use has been made of Breshenham's algorithm 1, which is extremely fast. The reason for this is that the normal incremental algorithm requires dy/dx to be calculated, the result of which must be stored as a binary fraction to preserve accuracy. Division and fraction handling are both extremely time consuming and so Bresenham's algorithm uses only integer addition, subtraction and multiplication by two (which is achieved quickly by a logical shift left).

To use the routine, first store the start and end points in the area LDATA in the following format:

LDATA : X1
LDATA+2 : Y1
LDATA+4 : X2
LDATA+6 : Y2

where the Microsoft notation is LINE (X1,Y1)-(X2,Y2). If the line is to be PSET, then store the value of the background colour in the variable BKGCOL (black = 0, green or white = 1 in either SCREEN 1,0 or 1,1). Storing the value of the foreground colour in BKGCOL will cause the line to be PRESET. Then call the routine with BSR STLINE. All registers used are preserved.

Note that all the coordinates are 16 bit values in order that negative numbers are handled correctly — a 16 bit register should therefore be used to effect the transfer.

Note also that X2 *must* be greater than X1 — ie a line is drawn from left to right. This speeds up the routine. Failure to observe this notation could produce strange results.

Vertical lines are extremely fast. Use can be made of the fact that once the position of a pixel has been determined, it does not change horizontally and so all that is necessary is to move the set point by 32 bytes each time. To draw a vertical line, first store the upper Y coordinate in location YU and the lower Y coordinate in YL, where YL YU. The X coordinate is stored in XVERT. The routine is then called with BSR VLINE. Again, all registers are preserved and the stacks are unaltered. Note that the vertical line is always drawn from top to bottom and so YU should always contain the Y coordinate which has the smaller value.

Both routines take the top left of the screen as (0,0). Although this is an unbelievably stupid notation, presumably conceived by some particularly asinine wally at Microsoft, it has been retained for the sake of familiarity (it also makes calculations that little bit easier).

Subroutines

The subroutine WBYTE returns the address of the byte containing the graphics coordinate stored as (XCORD,YCORD) in location BYTE using the relationship:

BYTE = SSTART + (32*YCORD) + (XCORD/8)

where SSTART is the base address of the screen. This is stored by Basic in location BA (hex) so the routine produces the correct result from any graphics start page (eg PMODE 4,1 ... PMODE 4,2 etc) irrespective of whether or not discs are connected.

The subroutine WPIXEL determines which pixel within BYTE is being referenced and returns the mask in the B register (ie if you imagine the B register mapped onto BYTE, then there will be a 1 at the position of the pixel and a 0 everywhere else).

One further point — neither STLINE nor VLINE check to see if the line being drawn is actually on the screen. It would therefore be quite possible to draw a line from (0,0) to (65535,655535) although if you try it you will almost certainly crash your Dragon — beware!

Finally, if anyone has any problems or wants anything explaining further, write to me at 83 Plain Spot Rd., Brinsley, Nottingham. NG16 5BG enclosing a stamped SAE.

References

1. Algorithm for Computer Control of Digital Plotter, J.E. Bresenham. *IBM Systems Journal* 4(1), 1965, pp25-30.


```
*******************************
*                             *
* PROGRAM 'lines'             *
*                             *
* LINE COMMAND FOR THE DRAGON *
*                             *
* 19-8-87                     *
*                             *
*******************************
*
        ORG     10000
*
STLINE  PSHS    X,Y,D,U
        LDX     $BA
        STX     SSTART
        LDA     #01
        STA     IORD1
        STA     IORD2
        STA     FLAG1
        LDD     LDATA+4
        LDX     #LDATA+2
        LDY     #LDATA+6
        LDD     ,X
        CMPD    ,Y
        BGE     SWAPY
        LDD     ,Y
        SUBD    ,X
        STD     DY
        BRA     CONT1
SWAPY   SUBD    ,Y
        STD     DY
        LDA     #00
        STA     FLAG1
        LDD     DY
CONT1   CMPD    DX
        BHI     YINC
        LDX     #XCORD+1
        LDY     #YCORD+1
        LDA     LDATA+5
        STA     FINISH
```

```
         SUBD  LDATA
         STD   DX
         BEQ   LEAVE1
         LDA   #-1
         STA   IORD2
LEAVE1   BRA   INIT
YINC     LDX   #YCORD+1
         LDY   #XCORD+1
         LDA   LDATA+7
         STA   FINISH
         LDD   DY
         LDU   DX
         STU   DY
         STD   DX
         LDA   FLAG1
         CMPA  #01
         BEQ   INIT
         LDA   #-1
         STA   IORD1
INIT     LDD   DY
         LSLB
         ROLA
         STD   INCR1
         SUBD  DX
         STD   DEE
         LDD   DY
         SUBD  DX
         LSLB
         ROLA
         STD   INCR2
         LDA   LDATA+1
         STA   XCORD+1
         LDA   LDATA+3
         STA   YCORD+1
         LBSR  WBYTE
         LBSR  WPIXEL
         LDU   BYTE
         LDA   BKGCOL
         CMPA  #00
         BEQ   BLACK6
         COM   ,U
         ORB   ,U
         COMB
         BRA   WHITE6
BLACK6   ORB   ,U
WHITE6   STB   ,U
LOOP1A   LDU   -1,X
         LDA   IORD1
         LEAU  A,U
         STU   -1,X
         LDD   DEE
         CMPD  #0000
         BGE   ELSE
         ADDD  INCR1
         STD   DEE
         BRA   NEXT
ELSE     ADDD  INCR2
         STD   DEE
         LDU   -1,Y
         LDA   FLAG1
         CMPA  #01
         LDA   IORD2
         LEAU  A,U
         STU   -1,Y
NEXT     LBSR  WBYTE
         LBSR  WPIXEL
         LDU   BYTE
         LDA   BKGCOL
         CMPA  #00
         BEQ   BLACK5
         COM   ,U
         ORB   ,U
         COMB
         BRA   WHITE5
BLACK5   ORB   ,U
WHITE5   STB   ,U
         LDA   ,X
         CMPA  FINISH
         BEQ   LINEND
         BRA   LOOP1A

LINEND   PULS  X,Y,D,U,PC
*
*================================*
* DRAW A VERTICAL LINE FROM      *
* YU(UPPER) TO YL(LOWER)         *
*================================*
*
VLINE    PSHS  D,Y
         LDD   $BA
         STD   SSTART
         LDD   YU
         STD   YCORD
         LDD   XVERT
         STD   XCORD
         BSR   WBYTE
         LBSR  WPIXEL
         STB   MASK
         LDD   BYTE
         STD   DX
         LDD   YL
         STD   YCORD
         BSR   WBYTE
         LDD   BYTE
         STD   DY
         LDY   DX
LINLP    LDB   MASK
         LDA   BKGCOL
         CMPA  #00
         BEQ   BLACK7
         COM   ,Y
         ORB   ,Y
         COMB
         BRA   WHITE7
BLACK7   ORB   ,Y
WHITE7   STB   ,Y
         LEAY  32,Y
         CMPY  DY
         BLO   LINLP
VLEND    PULS  D,Y,PC     ;RETURN
*
```

```
*******************************
*                             *
* VARIABLES & SUBROUTINES     *
*                             *
*******************************
*
MASK   RMB  1
DEE    RMB  2
DX     RMB  2
DY     RMB  2
INCR1  RMB  2
INCR2  RMB  2
FINISH RMB  1
SSTART RMB  2
IORD1  RMB  1
IORD2  RMB  1
FLAG1  RMB  1
XCORD  RMB  2
YCORD  RMB  2
*
BYTE   RMB  2
PIXEL  RMB  1
*
YU     RMB  2
YL     RMB  2
XVERT  RMB  2
*
LDATA  RMB  2         ;X1
       RMB  2         ;Y1
       RMB  2         ;X2
       RMB  2         ;Y2
*
BKGCOL RMB  1
*
*==============================*
* CALCULATE IN WHICH BYTE THE  *
* PIXEL TO BE SET/RESET LIES   *
*==============================*
*
WBYTE  PSHS D
       LDD  SSTART
       STD  BYTE
       LDA  YCORD+1
       LDB  #32
       MUL
       ADDD BYTE
       STD  BYTE
       LDB  XCORD+1
       LSRB
       LSRB
       LSRB
       SEX
       ADDD BYTE
       STD  BYTE
       PULS D,PC      ;RETURN
*
*==============================*
* CALCULATE WHICH PIXEL AND    *
* LOAD B REGISTER WITH MASK    *
*==============================*
*
WPIXEL PSHS A
       LDA  XCORD+1
       ANDA #$07      ;00000111
       STA  PIXEL
       CMPA #00
       BHI  NEXT1
       LDB  #$80      ;10000000
       PULS A,PC
NEXT1  CMPA #01
       BHI  NEXT2
       LDB  #$40      ;01000000
       PULS A,PC
NEXT2  CMPA #02
       BHI  NEXT3
       LDB  #$20      ;00100000
       PULS A,PC
NEXT3  CMPA #03
       BHI  NEXT4
       LDB  #$10      ;00010000

       PULS A,PC
NEXT4  CMPA #04
       BHI  NEXT5
       LDB  #$08      ;00001000
       PULS A,PC
NEXT5  CMPA #05
       BHI  NEXT6
       LDB  #$04      ;00000100
       PULS A,PC
NEXT6  CMPA #06
       BHI  NEXT7
       LDB  #$02      ;00000010
       PULS A,PC
NEXT7  LDB  #$01      ;00000001
       PULS A,PC
*
```

Word processor applications

***Roger Merrick** finds that there is more to WP than words*

DOES the subject of word processor applications need an article, you may ask? Surely with a word processor, you process words. Depending on whether you are at home or at work, this may involve writing letters, essays, novels, articles for computer magazines, reports and so on.

Well, there are more applications for a word processor than may immediately meet the eye.

OS-9 users will be well aware that *Stylo* can be used to prepare program text for Basic09, ASM, C, Pascal and so on. In a similar way, Dragon system word processors can be used to prepare Basic programs.

Load as Basic

If working with cassette input/output, a word processor like *Minitext* can be used to prepare an ascii file that can subsequently be loaded directly as Basic.

The obvious disadvantage of preparing Basic in this way is that it is not interactive, ie programs cannot be run from within the word processor. Cassette I/O makes the process of saving the word procdssor files, and loading and test running the Basic a lengthy and frustrating one.

However, there are advantages if the system is used correctly. It is probably not helpful to write from scratch a Basic program within a word procesor, but when modification of an existing program or programs is required, then the global/selective search and replace facilities of a word processor come into their own.

Merging one (program) file with another is an inbuilt feature of most word processors. The files will be loaded sequentially in memory, up to the limit of workspace. There are no worries about shared line numbers accidentally overwriting each other.

Block move

Block move operations can be used to restructure programs. *Minitext* contains RESEQUENCE commands that automatically correct line numbers.

With justification off, a line can be appended to a previous line to optimise program operation.

Files can be saved to disc, speeding the whole process of program development, but the question of getting the DATA files into memory as Basic is not straightforward. At present I load from cassette a valid Basic file heading into the world processor, then I merge the DATA file from disc, and finally save to cassette. The resulting file can be loaded as Basic. There is certainly a better way, and I would like to hear it.

Disc doctoring

If the word processor allows any form of mail merging, build a mailmerge file of the Basic tokens, two byte tokens first, single byte tokens subsequently. In the same order, build an equivalent file fo the ascii text Basic words.

If a Basic program is accidently killed on a disc, use DragonDOS's SREAD command to read and display the sectors until the remains of the required program are located. The brighter folk can read the directory content and interpret the information to locate the lost file directly, sometimes. Having located the remaining sectors of the lost file, they should be read into an array in memory, and then written to a data file on a disk. This data file can then be read into the word processor, and the mailmerge facility can be used to replace the Basic tokens with ascii text equivalents, producing a readable version of the program and facilitating the reconstruction of any lost/garbled areas. The resulting file can be saved as described for Basic files above.

The reason a mailmerge facility has to be used rather than search and replace is of course that the characters used as Basic tokens are not available from the Dragon keyboard. Some word processors may allow the search string to be defined in terms of CHR$(nnn)+CHR$(mmm), which would allow the tokens to be searched for.

Disassembly

In a similar way, a file of assembler mnemonics and their decimal equivalents can be constructed. Working preferably from a disc based file, a machine code file can be read as data into the wordprocessor and mailmerged with the assembler mnemonics file. Working from a disc file is likely to be required as word processors cannot be moved around in memory, and generally reserve all availble memory for their own use.

Database

Many of the databases that I have seen for the Dragon are highly restrictive about the way they can be searched. How irritating I find it, to search a database of names and addresses, to find that I must specify an entire search string, not a substring. If:

John Williams

is a name, and I search for Williams, depending on the database, I may get a 'no record found' type of report. Perhaps I wish to prepare some information based on the location of addressees in the database, by collating all those in certain postcode areas. Again, many databases will not cooperate with a desire to collate all postcodes in area B16, if the postcode field is 7 characters long. If I'm lucky, the database may automatically fill in wild cards for the rest of the field.

But with a little self imposed discipline, data can be structured in a word processor in such a way that the word processor's search facility can be used to find every occurrance of particular strings. Of course, this is only a one-dimensioned search, but many of the Dragon databases I've seen are only one-dimensioned anyway.

The advantage of the word processor is that a search can be made of any search string that can be entered from the keyboard, from one letter to the word processor's search string buffer limit (varies - *Minitext*'s is 254 characters). A disadvantage of course is that data has to be loaded in rather large blocks, whereas a database should read on a string by string basis. In prac tice, many Dragon databases seem to load database information in big blocks anyway.

Another use I made of the word processor was to construct a yearly diary. Twelve files, each named for a month of the year, and initially consisting of dates 1 to 28/29/30/31 depending on the month, with the revelvant days of the week appended (using block copy and repeatedly saving text under different filenames to minimise typing).

I can then enter data as required, and use the word processor's search facility to find any coming appointments/memos.

Accounts

Not many Dragon word processors have the facility to permit alignment and summation of figures. *Minitext* is one, but if this feature exists then simple accounts work can be carried out within the word processor.

Conclusion

A word processor is a powerful and flexible piece of software, which lends itself to applications wider than simple preparation of text. With the limitations of DragonDOS, and the difficulties of transferring files produced within one application being loaded into another, a powerful word processor can perform many of the functions of simple text processing software, such as databases, as well or better.

Show us an alternative

Ken G. Smith makes a great journey into the Black Country to find the legendary Einstein, and maybe Dragons.

THE question I kept asking myself was "Is it worth a round trip of four hundred miles in order to attend the first ever Alternative Computer Show?". I have long believed that the way forward for us was to form an alliance with users of another of those machines that the mainstream of the computer trade ignores. When Emsoft announced their event to be held in Birmingham, I felt that I had to go. The only problem was that journey. As things turned out even the morning fog did not get in my way, and I arrived in Birmingham only three and a half hours after I set out.

I have to say the directions in the advertising left a lot to be desired. Junction 6 on the M6 may be easy enough to find, but from there it was assumed that we would all know our way. It would not have taken much to post a few "to the show" signs, or at least include in the advert "follow signs for Aston Hall until ... ". I spoke to several people who had spent half an hour or more trying to find the Aston Villa Sports and Leisure Centre. Those who did find it were rewarded with a great show with over forty exhibitors, of which unfortunately only four had Dragon connections. I must admit that I did find this a bit disappointing, only Oric and Texas Instruments were less well served. However, I think that this was partly due to the unfortunate timing, the Weston show being only three weeks away. The event was originally conceived as a Tatung Einstein show, and only recently was it decided to include other machines. This also meant that there was less time to recruit exhibitors from other brands.

Since the mainstream computer press started treating the Dragon as if it was a four letter word, we Dragon users became a bit isolated and some tend to think that we are the only ones with enough loyalty and devotion to our machine to keep it going through thick and thin. Well, if you thought that, then you are wrong. The TI99/4, for instance, ceased production long before the Dragon, but it still has an active, if small, user base and Emsoft, the organiser of the show, is a software house dealing solely with Einstein programs. The problem we all face is our shrinking numbers. Alone not one of us is capable of supporting a show of any size, but together we can manage something big enough to attract suppliers of monitors and printers, etc. The Aston show had about five stands allocated to such people. Cut price floppy discs were everywhere. Only when the 6809 Shows were really packing them in some three or four years ago did such people attend.

Several groups attended, even regional ones for the Einstein, one each for the TI99/4 and Lynx. Even the Jupiter Ace managed a group. The National Dragon User Group, however, were conspicuous by their absence. Of the four Dragon software dealers, not one found any cause for complaint. John Penn said that even if he had not covered his costs (he thought he had) it would still have been worthwhile attending. Prestons and Computape reported a reasonable trade, though not as good as at Dragon shows. The fourth was a real blast from the past: Capri Marketing seemed to sink into obscurity some years ago, so it was a surprise to see them at the Aston Show. What was even more surprising was that they are still selling Dragon software, amongst other things, of course.

Everyone attending the show got a free copy of *Micro Computer Mart*, a fortnightly magazine mainly designed as a vehicle for advertisers, but also containing some interesting articles. The free ads for readers make it attractive and it is a relief to find a non-aligned magazine that does not treat Dragoners as if they were brain dead.

I found at least one new friend at the show, and stayed longer than I had intended. The organisers are already planning another, bigger show for the same time next year. This gives us plenty of time to arrange our diaries in order to accommodate the event. Alone our shows can only get smaller. Combining with others will enable us to continue longer and make more people aware of the fact that we are still here.

*I think an all-micro show such as Emsoft are hoping to establish is an excellent idea and one which will help to give users of older micros more contact and staying power. This is the first year that Emsoft have thought of drawing in other machines, and the timing, as Ken says, and as I have pointed out to a number of interested parties, was indeed unfortunate, so much so that a rumour was started that **Dragon User** was 'boycotting' the show. This was far from the case, but we had an obligation to put most of our support behind the Weston show, to which many Dragon dealers had already committed their resources. In future years I have great hopes that the Alternative Micro show will prove a suitable forum for the Dragon, and we will support them if they support us.*

Breaking out in Chains

J R Sutcliffe tackles an old command.

I expect many people have tried the CHAIN command when they first got their disc drive, and like me they gave up trying when their programs crashed.

It is a pity, because if you can get it to work, it allows the possibility of 20K of data with 100K of program in a 32K Dragon, with hires screen and DOS, and space to spare. The limit of the program size is the size of the disc. My experience with this command is the result of hours of experimentation trying to get a quart into a pint pot, ie converting a program which PCLEARED1 to run under Basic 42.

1) Write a module (PROG1) which clears string space, and dimensions all strings and variables, the last statement being CHAIN"PROG2".

This module may not be very large. One of the foibles of the Dragon CHAIN command is that though you can CHAIN smaller programs from larger ones, if you try the other way round, they crash. Pad out all modules with 256 byte REM statements, up to, for example, 4K. If any module gets to more than 4K, pad out the others till they are all equal. THIS IS VERY IMPORTANT.

2) Write the menu and other modules PROG2,3,4,n. If when writing these, another string or variable is required, edit them into PROG1.

3) Modules can chain any other module in any order. All data and variables are maintained, only the program is changed.

4) Sometimes the CHAIN command takes a long time (a few seconds). In order to prevent the impression of a system crash, print something on the screen before you chain the next module. For example:

```
100 CLS:PRINT"EDIT MODULE PLEASE
WAIT":CHAIN"EDITMOD"
```

5) It is recommended in the manuals to do a string compression FF=FRE$ at the start of each module. This adds to the time taken by 4, but it is perfectly acceptable to print instructions on the screen, and then do it. For example:

```
10 CLS:PRINT"INSTRUCTIONS FOR
EDIT":FF=FRE$
20 REST OF PROGRAM
```

These delays are the penalties which we have to pay for having so much apparent memory. CHAIN will not crash as long as all modules are within 255 bytes of each other, and works faster when they are the same. The command CHAIN"PROG" line is probably best used when expanding an existing program. If starting from scratch, less editing of other modules is required after they are written if all modules start at the beginning, or make use of a variable which is passed. For example:

```
10 IF N=1 THEN 100 ELSE IF N=2 THEN
200 ELSE 20
20 Rest of program
```

Since only one module is worked on, and saved at a time, development can be much easier. Sloppy development only loses one module if there is a crash.

Now I have outlined the way of using CHAIN, it would be nice to find a rash of new programs making use of the new found space. Spread sheets, CAM CAD, PCB design, who says you need 256K of memory to run useful programs?

Dragonsword!

***Paul Grade** gets his inspiration from a fruit.*

DO any of you remember those stories we used to read in just about every paper and magazine, the ones which used to rave on about the "Age of the Computer" and how only the current generation of kids would be able to understand it? The stories aren't quite so frequent now, and it's ages since I last read one about "Superbrat age 9 writes Megagame before breakfast and signs million pound contract before lunch", but somehow the belief still lurks that anyone old enough to buy a drink is too old to understand computers.

I often wonder how such myths get started, because if the members of the NDUG are anything to go by, this one couldn't be more wrong. We have about five people under the age of twenty who are adequate programmers, and a couple of those are what I would consider to be professional class, but we've about the same number who are over seventy, and while they may like to think of themselves as youngsters, I don't think they quite fit the image that the media was trying so hard to push! Most of the 'professionals' in the group come into the thirty to fifty age range, and although most of them aren't writing commercially for the Dragon, they earn their living writing for far more complex systems, so where does all this rubbish about Superbrat programmers come from? And why did it start at all?

Personally, I think it began as an attempt by computer manufacturers to convince parents that their beloved brats would end up unemployable and begging on street corners if they were deprived of a ZX80 or an Atari 400 on which to learn these new mysterious skills, which alone could set them on the road to fame, fortune and an Amex Gold card.

Convincing tale

There's nothing new about that kind of sale pitch ... it's a well-proven fact that most parents will buy their kids anything they think 'will give them a better start in life', even if they can't really afford it ... you've only to look at the pre-Christmas TV advertising to see how well it works. Of course, it is a double edged line too, because most brats are bright enough to play along with it, and when they get their new toy (sorry, computer), they usually make a pretty good job of convincing parents that they are brilliant programmers, even if their ability really only extends to loading in the latest Wallysoft Hypergame (converted for PC from the original Vic 20 hit!).

So what am I telling you this for? Not, as you expected, in an attempt to bore you to death, but in the hope that someone out there on the other side of the keyboard might be able to tell me the answer to the problem which is steadily killing the home computer scene. If the brats aren't learning to program, or to write about the subject, and the 'professionals' are all engaged in making money on the far more profitable 'business' market, where is the home computer material going to come from?

No Kidding

I don't think I'm risking Editorial Disagreement when I say that even user groups and magazines are short of informative articles on hardware and software, and software distributors have reached the stage where they've virtually nothing new to distribute. (*Far from editorial disagreement. Dragoners are lucky, because we have some very good people, and a well-informed following, who come up with a steady stream of information and comment, even if it isn't a flood. One magazine of my acquaintance is finding it so difficult to find any authoritive information on hardware that it is having to take insults with a smile. This is because the younger users aren't learning fast enough, and the experienced ones are all concentrating on making a living. However, your comments on software distributors may be met with a howl of disagreement as, after a lean time when everybody was mourning the lack of colour-packaged mega-conversions, they've got their act together and are producing new, inexpensive material steadily.*

It would help if schools did more towards helping kids to learn to use computers properly, but there doesn't seem much chance of that happening. Although there are exceptions, most seem to rely on teaching the kids which keys to press in order to run ready-written material, which is like teaching them to turn on a light and calling it electrical engineering.

Fortunately, there are still a few kids who don't believe all they're told at school, and some of them hack away at their computer at home until they can make it do what they want without having to rely on someone else's ability, but they are few, and they usually get little or no encouragement, and several I know have given up because they are always being told to 'stop messing about on that computer and do your homework', be people who can't see that exam passes merely demonstrate the ability to pass exams, not aptitude at anything useful.

What can you do?

So what can you do? Well, if you're still at school, you could try teaching yourself, rather than kidding yourself that 'Computer Studies' will teach you everything useful. If you're more ancient, you could stop complaining that 'I'm too old to learn that sort of thing' and have a go at it. (*My father-in-law is 83 and he got his first computer last year. He's teaching himself to program. He says the evening classes help a bit, but what you really learn from is sitting down and doing it. Ed.*) You could then pass on what you know to others of all ages, so that they have some incentive to learn. A little help and encouragement can do a lot of good.

Computing isn't a spectator sport. Sitting around playing with some else's program is about as instructive as watching *Neighbours*. Learn to write your own programs, learn to make, repair, and adapt your own hardware if necessary, and you'll find the whole thing becomes a lot more interesting.

I'm not suggesting that you shouldn't buy software, but that you should reach the stage where, if the program you need doesn't exist, you can write it for yourself, and if your Dragon expires in a cloud of smoke and a screen full of garbage, you at least know where to look, even if you can't fix it yourself in the final eventuality.

Yes, I know, if I'm so good, why don't I do more about it myself? Fair question, but I don't really know what more I can do. Running NDUG takes over sixty hours a week, I've repaired more Dragons and assorted peripherals than I care to remember, I do my best to answer around fifty or so queries a week, and write the odd (or downright peculiar) page for *D. User* each month. Generally, I try to needle people into realising that owning a computer is not quite like owning a video recorder. It's a two way machine. It may not make me very popular, but it gets better results, and it hopefully stings a few people out of their complacency long enough for them to start thinking about what they can do with their own resources, which in some cases is a whole new experience for them!

Tell me if I'm wrong

So that's why I don't run a software company, or do some of the other things I would like to do, because I don't have any time left to do them in! Anyway, you own a Dragon too, don't you, so why should I deprive you of your chance of fame or notoriety?

Yes, I do get tired of keeping on at people, but someone has to do it, and I got elected a long time ago!

By the way, if you are still at school and think I'm being unfair to your or your school, and that you really are a good programmer, and they really are teaching you something useful, write and tell us, I'd be pleased to be proved wrong. It isn't really that I eat people (I'm on a diet).

By the way, every letter that I've had in response to this column has been in agreement with my comments, and we'd like to hear from those who DON'T agree as well! (*We did get one, Paul. See Letters Page!*).

Expert's Arcade Arena

Write to 'The Expert' at Dragon User
49 Alexandra Road
Hounslow, Middlesex TW3 4HP

HELLO! I know you're dying to know what's in the column this month, so I'll waste no time. It's nice to be back on my throne again, and thanks to my minions for keeping it warm while I was away. Heavens! Hasn't there been a lot happening recently! Quickbeam disappearing and then Pamcomms, Radio One has gone stereo, Neighbours has grown ever more nauseating, Dragonfire and Orange have grown and isn't Mandragore good, but the big news is that Dragon User has moved (*You can see how long he's been away - Ed.*). I'm a bit unsure about the triffid wallpaper on the walls of my new office! Anyway, don't forget to send all those letters, postcards, car-keys, cigarettes, bottles of gin, proposals of marriage and the rest of the gifts to the right place. (*Gin to the Editorial office, all the rest to the Basement*).

So, to business, and in this month's column you'll find all the trivia and jokes and pokes and cheats that've always found on these pages. Pokes this month include *Mudpies* (does anyone know of an infinite lives poke for *Mudpies*? Extra lives seem to disappeear when you play the 'Mudslinger' screen) *Cashman* and *Guardian Angel*, for which I send my thanks to Michael Dunn - Hello, Michael, thanks - and there'll be more from him in the near future. Also this month is a feature in which, in the style of other great geniuses who answer questions from their beloved public, I'm going to call "Expert on the Spot". Send your questions and queries to EOTS, however small and trivial or drastically important for the survival of the human race, and I'll answer them as soon as you can say 'contrafibularities' in Portuguese.

Pokes

Mudpies ... (C) 20 POKE 29629,x (127)
Cashman ... (C) 20 POKE 28703,x (255)
Guardian Angel ... POKE 10802,x (255)

Here's the first problem for the Expert on the Spot, from Keith Porteous: "I have now finished *Beanstalker* - this guy's got a superhuman patience barrier - and have been making my own screens with the editor, but what puzzles me is whether the screens I design will be saved to tape." Familiar to us is the "5,R,E,G,I,S,T,E,R, ENTER,N,ENTER/X/8" cheat for *Beanstalker* but, as Keith says, you can't save the screens. To save the screens, you do in fact need to splash a bit more cash and buy an add-on, known as *Beanpatch*.

As far as I'm concerned, to be asked to give away yet more of your hard-earned pennies for just a few nibbles of program seemed a little too much, so I haven't invested in the tape. I gather *Beanstalker* is now supplied by Orange Software (although their version may be different to the original Microvision version - if anyone out there knows, why not get in touch and let me know?), and *Beanpatch* is also available for £1.00. By the way, all this information can be found on the Newsdesk of the May 1988 *Dragon User*. Orange is quite a busy company these days, and keeps releasing new products, so please support them and help them establish themselves in the market-place. We can't afford to lose any more Microdeals or Quickbeams and Orange look set to become a vital link in the future.

Ye-haw

Now then, a quickie from Eric Hornby. He wants to know why his 'doubtfully acquired' copy of *Module Man* crashes every time it loads. Are you sure it's not a Tandy version? Otherwise, my copy always works, so quite bluntly, mate, it looks as if it's duff. Don't buy from the cowboys in future - and take a slap on the wrist. Here's one for you lot to work on, also from Keith: How do you get off the last screen of *Catacomb Crisis*? No-one has sent me this game yet, so I still can't help. Are my hints really too subtle? More EOTS next year. (That's tomorrow, dears, in case that's still too subtle.)

Right, now onto a letter from someone who's obviously keen to make new friends, as they begin "I hereby claim the *Expert's Arcade Arena* to be the most unreliable column in the entire history of *Dragon User*." (*Too right. Who wants predictability? Apart from me, of course. Ed.*) Well, thank you for your kind comments Ola Eldoy, for words which will stay with me forever, amongst the other precious memories of this, the high point of my career. Ola also makes amends by providing the cheat procedure for *Rolaball* (cheers all round). My thanks to Ola, who was actually writing on behalf of the Dragon Compuerklubb (the Scandinavian user group), and Hello to all of you over there in Norway.

Rola-ball Cheat

Press the '&' key during play. The code is 'DESTINY'

By the way, if anyone managed to miss MacGowan Consultants' advert for *Monster Mine* in the September DU, then go back and cast your eyes over it. It makes for a welcome change, in fact I wish I could send them a prize for plain, cold honesty. Maybe this could start a craze among advertisers and we'll see them fighting each other by criticising their own products! - and why not, it worked for British Rail and Mello? (*Who???.*) PS, how about a copy, Mac?

Well, it seems that that's just about all there's room for in the mag for this month (can you see any blank pages?) (*Don't say such things! This is tempting fate ...*), so I'd better say Cheerio sooner or later (and believe me, I'd sooner it was later than sooner). Firstly, however, let me tell you that you'd better read next month's column because teams of volunteers, even as I speak, are counting your votes in the software survey and if the round-the-clock schedule goes according to plan, then the results should be ready for me to go public next month. May the Good Lord help you sleep between now and then. Bye and Merry Christmas.

Communication

Problem: Wanted: Dragon's Claw and Snap Camera.
Name: T. Glickman
Address: 15 Epping Drive, Sale, Cheshire M33 5LR

Adventure: Bedlam (Radio Shack)
Problem: Can only get green key and recover from lobotomy.
Name: Kevin Barrett
Address: c/o BIH, Sumburgh Airport, Virkie, Shetland.

Problem: I lost my instructions for Dragon Composer and I can't operate it properly. Can anyone help?
Name: Colin Dickie
Address: 2 Greenbank Road, Cullivoe Yell, Shetland ZE2 9BZ.

Problem: Wanted, Elite-Calc speadsheet, also Dynafast compiler. Both for DragonDOS.
Name: T C Hanson.
Address: Tel: 0273 594654.

Communications

Write down your problem on the coupon below (make it as brief and legible as possible) together with your name and address and send it to Communication, 49 Alexandra Road, Hounslow, Middx TW3 4HP.

Problem . . . .

. . . .

. . . .

. . . .

Name . . . .

Address . . . .

. . . .

. . . .

Write: ADVENTURE

Pete Gerrard slips naturally into a role

NEXT door's hellhound seems to have survived the postal strike, judging by the phenomenal amount of noise it was making only the other day. The cause of this noise is not known, although I note with some trepidation that there is a large pile of scaffolding outside the nearby public house, while no workmen have been seen for two days. Mind you, precious little can be seen at all today, it is ideal role playing weather.

There is a liberal coating of frost covering everything, it is extremely misty, thus rendering the cemetery on the other side of the road invisible, and it is very, very cold. Cemetery? Yes possums, a cemetery. Distinctly spooky it can seem at times as well, like on dark frosty nights when a low mist covers the ground and only the tops of the gravestones can be made out in the eerie half-light. Someone once remarked that next door's hellhound was barking "fit to raise the dead". I sincerely hope that it doesn't!

In last month's issue I gave a brief introduction to role playing games, but there are many other aspects of this fascinating trend in adventures that have still to be looked at. Quite a few traditional adventures have had characters in them, or 'pets' to give them an official name, but these characters are to be found in abundance in role playing games, and just as early adventures all tried to beat each other by having three trillion locations, so modern RPGs are wandering along a similar road by trying to cram in as many characters as possible into a single game.

I notice K. Hunter's new adventure, *The Curse of Camarc*, is strong on characters, and includes such delights as an elf, a hermit, and a basilisk. Over one hundred locations as well, which can't be bad, and text only to boot. Now this would be a good person to write a role playing game, and push the dear old Dragon even closer to the limits.

Role playing games

Characters in RPGs can usually be divided up into good or bad. The good ones will, at the very worst, ignore you, but others will undoubtedly help you to varying degrees. Since most RPGs are, at the moment anyway, merely slight variations on the tradition Dungeons and Dragons theme, something that we'll come back to later, the good guys are almost invariably along the lines of wizards and elves. The bad guys will, at the very best, only put up a timid fight, but others will be doing their utmost to destroy either you or members of your party. Again the Dungeons and Dragons theme is strongly prevalent, and the usual assortment of orcs and balrogs romp around with other, newer, upstarts.

Bartering is usually to be found somewhere along the way, and the status of the character that you're bartering with, whether they be good or bad, will determine many things. Not least of these will be to see if they can be bothered to deal with you in the first place. They might instead push you away and ignore you, but eventually you'll get round to doing a spot of bartering with someone at some time.

Careful with goats

There's an important rule to obey when bartering in an RPG: never be ridiculous. That is, if an object costs, say, 500 groats, don't go offering 10 for it. The character that you're dealing with might be highly unscrupulous, and at your rank bad manners produce a meat cleaver from somewhere and effectively end the game. You might be lucky and escape with a warning, but then it will probably take you many moons before you're allowed to attempt to barter again. Always try something reasonable, like 400 groats, and take it from there. You'll probably end up paying around 450 for whatever the item might be, and a saving of 50 groats on the original price can't be bad. 10 per cent off, well worth bartering.

Another rule applies just as much in RPGs as it does in adventures: always save your position before trying anything dangerous or foolish. Inns and taverns usually pop up in an RPG, and there you might be offered all sorts of delights to tempt your palate. Grog and mead are inevitably there in one form or another, but it's probably safer drinking water. I've played one RPG where if you try and have too much strong ale then you get drunk and cannot perform anything effectively. Reasonable enough, you might assume, but in this particular game it was not only the character who got drunk. So did the computer! At first I was convinced that the blessed thing had crashed, but as my character gradually recovered from the effects of his session so did the computer. An interesting experience.

This getting drunk in taverns lark is just one area where RPGs score over traditional adventures, because there seems to be a much closer link with reality. You get a more vivid impression of actually being there because your character gets hungry and thirsty, or tired, and needs to find food and water and possibly a bed for the night. I know that some adventures have tried to do this, but it never seems to work in the middle of an adventure game when you're trying to puzzle out how to get across a yawning chasm. There is less concentration on problem solving in RPGs, or so it seems to me anyway, and more effort is spent on convincing the player that they really are there, trying to work out how much money they've got and whether they can afford an expensive single room or a cheaper communal one and thus run the risk of being robbed in the night.

That is not to say that problems don't exist, because they most certainly do, and in great quantities as well. It is just that the problems are, generally speaking, of a different nature from the standard "insert yellow card in slot and pull lever" variety beloved of the adventure game writer. The problem of a room for the night, for instance, as mentioned in the previous paragraph. Checking in at the wrong place could be a potential disaster, although it is something that can be done rather than a problem that prevents you from getting any further on in the game until it is solved. Therein, methinks, lies another great virtue of RPGs: there's always something to envisage doing, you can always have a roam around and see what lies beyond the next corner, you're not stuck in front of a yawning chasm without a clue as to how to get across.

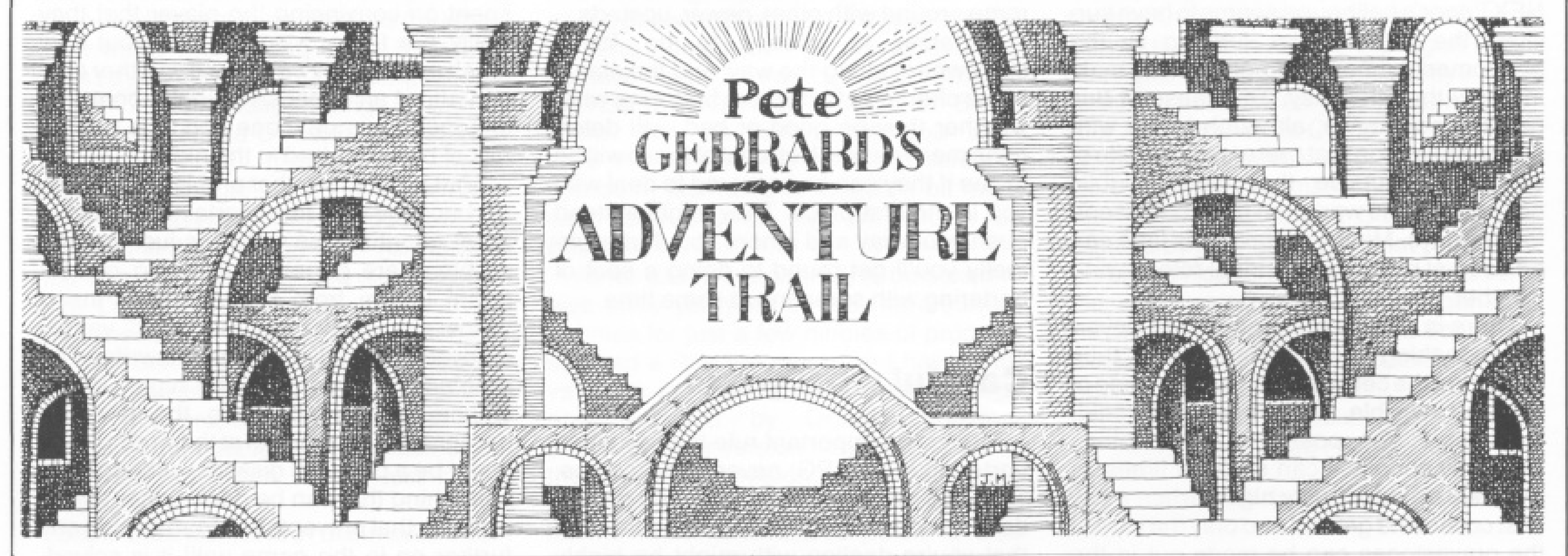
Time marches on

RPGs often tend to play themselves as well. That is, if left to their own devices and the player doesn't press any keys and initiate any actions, the characters within the game will roam around of their own accord, time will pass, and various events dependent on time will also happen. Inns will open and close, and if you fail to get a bed for the night and have to sleep out in the open then that's your look-out. Once again, this sort of thing has been tried in more traditional adventures, although it always seems unfair to me that time marches merrily on while you're STILL working out how to get across that wretched chasm.

But here we come back to what I think is the one major problem facing RPGs at present, and it was a situation that early adventure games went through and to some extent are still facing themselves. Just as early adventures seemed to be nothing but copies of the original *Colossal Cave* (and I was as guilty as anyone else when writing them) so nowadays do many of the more popular RPGs all look remarkably the same. They're not the same game, you're not in front of the same situations all the time, but there is a clear resemblance from one to the other. All the comments made so far about bartering, good guys and bad guys, taverns and inns, could apply to any one of a hundred RPGs, and it is to be hoped that someone (you?) with a bit of ingenuity can manage to break out of the mould and come up with something completely different. We might look at a few ideas next time around, but

first, the news.

Our beloved editor hath spoken, and foolish be the man who ignores her word. So, from next month onwards, there won't be two separate columns as there is be the moment, but instead there'll be one big column. This makes sense, because there is often a cross-over from one to the other, items that appear in the *Adventure Trail* could equally as well appear here, or vice versa, so from next month it'll all be lumped together but probably still called the *Adventure Trail*. Now here's where you come in. If there's anything you'd like to see in this new section, please write and let me know and I'll do my best to incorporate the cleaner, wittier, better, suggestions. What you get will thus, to a large extent, depend on what you want, so please put pen to paper or finger to keyboard and drop us a line. Oh yes, and if you've any suggestions for ingenious ways of quietly disposing of hellhounds, then I'd love to hear them. Until next month then.



I seem to have stirred up something of a Dragon's Lair with my comments in the September issue of *Dragon User* when suggesting the setting up of a nationwide Dragon adventure swap-shop. I received an extremely pleasant (and extremely long) letter from our old friend Jim Finley in Romford on the very topic: letter written on 2nd September 1988, answered personally as soon as possible, but only now reaching the glorious pages of our (and I must stress the OUR, meaning all of us) magazine. Such are deadlines, the bane of any editor's life, and any editor who can include in one of her letters the amazing comment "we can take some of our newly acquired snails and introduce them to the Welsh marches" can only be admired for putting up with these things.

Swapping or piracy?

But back to Jim's letter on the subject of swapping adventure tapes. It's good to see someone else putting Romford Brewery advertisements in their place, let me tell you. The company who put the bull in John Bull, no wonder it tastes nothing like beer. A minor digression. The topic of swapping tapes and "piracy" is, by its very nature, an awkward one. I don't agree with piracy any more than Jim does, I would never copy a tape or disk with the express purpose of making money for myself. I never have done and I hope I never will. On the other hand, if someone sends or gives me a piece of software that they have no other use for, then I will use it. If they have also kept a copy for themselves, on the premise that they might one day want to look at it again for whatever reason, then that is their business.

However, the crux of the matter lies with the vexed question of are we robbing people of money by copying and swapping games that are on the market and readily available through mail order schemes, and that by doing so are we then making potential new suppliers of Dragon software think twice about coming into the world of the Dragon? Let us, in the honourable British tradition, arrive at a compromise. If an adventure game that was once produced by a thriving company exists beyond the commercial life of that company and becomes impossible to obtain, then we can assume that unless the author decides otherwise we are free to make backup copies and swap them amongst our friends. Such a program, along with programs that are put into the world of public domain software, will then once more become available to the Dragon community. If that program is subsequently re-marketed by another, existing, company, then it no longer becomes one that can be swapped in the aforementioned manner.

To sum up, then. If a program is no longer being marketed in any way, shape or form, then we can make it available to all members of the Dragon community. If at any time it is either being marketed, or someone else takes up the marketing of it, then we can't. Sounds simple enough to me, and I trust that Jim and others will approve. So, if anyone has any Dragon adventures that are not being marketed, not being sold anywhere, and they would wish to share those adventures with other Dragon users, let us know. If I or anyone else subsequently discover that the game IS being marketed, the boys will be around to do unspeakable things. Failing that, Helen and her snails (assuming they survive the Welsh marches) will transform your life into a nightmare.

Classics

By the 'eck, that were all reet serious, 'appen. But it is a serious topic, so thanks to Jim and his letter I hope we've managed to get everything sorted out to the satisfaction of all concerned. And now, back to Dragon adventures, and two classics.

Yes, just two for the rest of this column. Readers with exceptionally keen eyesight will have noticed a map attached somewhere near this column, but we shall come back to that later. To begin with, a friendly letter from Andrew McBride (and a quick hello to another letter from Peter Hawes, the 16 year old adventure genius in September of 1987 who is now the 18 year old adventure star of 1989 — hello there) concerning the game The Final Mission.

The Final Mission

Now beware, ever vigilant readers, because what you are about to get is the COMPLEAT SOLUTION, as some famous fisherman or other might have spelt it. Yes indeed, the 100% this is what the final message reads solution. Minor drawback, you have to cheat. BUT, the game is written in Basic and so that makes matters simpler. I shall quote Andrew's letter to you:

"About two months ago (four now, postal strike still operative at the time — P.G.) I wrote to Incentive Software and asked for a solution sheet to *The Final Mission*. I received a solution sheet and thought it would be easy to complete. WRONG! For some reason there is no acorn, and you need to drop the acorn so to kill the spider, I have searched everywhere for this acorn but have had no success. About two days ago (months now — P.G.) I found out that the Ket adventures are indeed Basic. This is what you have to do:

Load the program as normal, when the computer goes 'click' (signalling that it has finished loading the program and is now loading the data) reset it, then type the following:

```
0 GOTO 9999
```

You then must EDIT 4210, press X and

delete out 'GOTO 7000'. All you have to do then is type RUN. The solution follows:"

This is Andrew's full solution now, so read as far as you need to (or dare) because I am not typing all this lot out backwards in the time honoured fashion: it would take forever! (*When is someone going to come up with a program that automatically transposes any quantity of copy into reverse?? Ed.*)

Not backwards!

(1) Get chair, examine chair, east, drop chair, stand on chair, break glass, east.

(2) North, get straw, east, south, drop ring, east.

(3) East, east, east, east, west, south, get garlic, west, south, west, south, get crowbar, west, south, get soap, north, east, north, east, north, east, north, east (boot should open trapdoor).

(4) Down, east, south, south, block gap, north, north, east, remove panel, east, north, drop garlic, south, east, get aerosol, west, south, east, east, south, south, west, west, up, wait, wait, wait, wait, wait, wait (you should hear a splintering sound), drop soap, down, north, north, north, west, west, south, south, south.

(5) Unlock door, drop key, south, kill monk (or escape), south, south, south, say "anagram", west, say "ga", east, south, get pendant, insert edgar, wear pendant, south, west, north, north, east, east, east, say "26,9", south, south, south, get carving, west, south, south, west (this is where you should drop the acorn), share aerosol, spray aerosol, down, south.

Use crowbar, drop crowbar, north, east, down, down, unchain man, get handle, up, up, south, south, get rubies, insert emeralds, south, south, show carving (make sure you are wearing the pendant with edgar in it!), south (the pendant will banish the demon back to hell), insert handle, south ..... THE END!

Andrew tells me that if you save or load the game you will not receive the full 100% and you will not get the final part of the secret message, however, having completed all three Ket adventures the secret message is EE (aarggrrumphhh! — sound of Andrew's voice being hushed up)

Solution sheets

Last bit of news from Andrew is that he has prepared an A4 booklet with hint sheets and solutions to *The Cricklewood Incident, Trekboer, Syzygy, Aquanaut 471, Demon Knight, Lost In Space, Mountains of Ket, Temple of Uran, The Return of the Ring* (and map). This costs £1, including postage, and can be obtained from Andrew McBride at 109 Main Street, Little Harrowden, Near Wellingborough, Northants, NN9 5BA. Good on you, ma boy!

Meanwhile, back at the map. *Total Eclipse* Universe 02 you will note, another *Dragon User* Exclusive (I hope!), brought to you courtesy of another Dragon stalwart, none less than Joe Brincat. I think he must spend his entire life playing this game. Anyway, he has mapped galaxy one of universe two of *Total Eclipse*, and a million blessings on him for doing so. (*The map has suffered a bit in the post, so bits of it are not perfectly legible, but we'll have a go at reproducing it, and if that doesn't work, we'll try again sometime. Ed.*) Saving a position is possible, reloading is not, which is a minor problem, so one can only stand back in amazement at the perseverance that this man has. As if producing this map were not enough, Joe offers help to anyone still stuck in Universe 01, and to quote "all they have to do is send an SAE and no money.

Wishful thinking

After all, this is a game & not a business transaction." Good lad. Joe lives at 73 Annunciation Street, Hamrun, Malta, for those who don't know. He tells me that Malta is soon to have a Motorola Company, who knows they might produce the Dragon Professional one day (wishful thinking there!), he says that lots of his friends would like to see the PC Convert working, and he wishes a Happy Christmas and Happy New Year to one and all. What more can I add? Have a fine '89 everybody, and let's keep the Dragon Motorola-ing (?!) along over the next twelve months.

Letters

This is your chance to air your views — send your tips, compliments and complaints to Letters 49 Alexandra Road, Hounslow, Middlesex TW3 4HP

Wordwrap workout

CONSIDERABLE time must have been spent by people writing text adventures or screen instructions for programs to ensure that words do not wrap round to the next line (li
ke this) Often, when I think I have it right, I find when the program is RUN that I am out by one character and the program has to be re-edited.

The following program segment and sub-routine will take care of the problem. I offer it in the hope that it may be useful to other Dragon users.

```
100 FOR A = 1 TO b
110 READ A$
120 GOSUB 3000
130 NEXT A
140 DATA xxxxxx , xxxxx , (etc.)
2995 REM — WRAP-ROUND
PREVENTION
SUBROUTINE
3000 Y = INSTR (1,A$," ")
3010 B$ = LEFT$ (A$,Y)
3020 IF LEN(B$) = 0 THEN
3100
3030 IF LEN(B$) =
31-POS(0) THEN? CHR$(13)
B$; : GOTO 3070
3040 L = LEN(B$)-1
3050 IF RIGHT$ (B$,1) = " "
THEN B$ = LEFT$ (B$,L)
3060 IF POS(0) = 0 THEN ?
B$; ELSE ? CHR$(32); B$;
3070 X = LEN(A$)-Y
3080 A$ = RIGHT$ (A$,X)
3090 IF A$ "" THEN 3000
```

OR

```
3090 IF LEN(A$) 0 THEN
3000
3100 RETURN
```

Notes: B in line 100 is the number of DATA items. The shorthand ? has been used for PRINT throughout. The semicolons (;) are essential. Strings of any length up to the maximum (255 characters) may be used. There MUST be a space after each DATA item before the comma, and there must be a space before the closing ' if the alternative below is used. Double line spacing can be obtained by shortening line 3030 to:

```
3030 IF LEN(B$) 31-POS(0)
THEN ? CHR$(13)
```

(To make the above program work, add the line 200 STOP)

Alternative lines 100-400:

```
100 W$ = "sentence 1 "
110 X$ = "sentence 2 " (etc.)

150 A$ = W$ : GOSUB 3000
160 A$ = X$ : GOSUB 3000
(etc.)
```

(The double quotes (") cannot be used within strings in the alternatives to lines 100 on)

Where several screenfuls of text are required, or where text is required at several points in a program, the sub-routine can be called on repeatedly.

With modification, the sub-routine could be used to print columns of text on a printer. (PRINT would become PRINT £ -2, POS(0) would become POS(-2) and the value 31 in line 3030 would be altered according to the column width required.)

R M Cashmore
Church Farm Cottage
Blaston
via Market Harborough
Leics LE16 8DE

Excerpt from an epic

THIS is a rag-bag of a letter and I hope you'll bear with me while I get a number of thoughts out of the way. (*1 ... 2 ... 3 ... 4 pages. This is no rag-bag, Jim, this is a paper mill!*)

I ws flattered when I found my letter in September's *Dragon User* and waited eagerly for the Armstrong benevolence. Nothing happened. Now, I have learned never to argue with an editor. No. That would clash with the doctrine of Divine Infallibility. The other possiblity is that you are waiting for me to say what I would like. May I say I would welcome anything that is not an arcade game or a text adventure and the more obscure and offbeat the better. (*Oh good*).

Jim Finlay
11 Fernden Way
Romford
RM7 9PJ

Cereal data

Gordon Lee unearths some startling factorials!

ON this page in the past we have made reference to 'cornflake' competitions. By this I mean those competitions - frequently found on the packets of breakfast cereals - in which it is necessary to arrange a list of desirable features (usually relating either to the prize on offer, or the cornflakes) in the order decided upon by a panel of judges. To be certain of sending in a winning line, how many attempts would you need to make?

Let's assume that there are only four features on the list. We would need four attempts to be sure of placing the first choice. This would leave three options remaining for second place, so we would need 3 *times* four tries to be certain of correctly placing the first two features. Proceeding in this way, it is clear that all four would need 4 x 3 x 2 x 1, or 24 attempts. This value is known as 'factorial' 4 and is written 4! - the exclamation mark being the symbol for factorials. A list of some of the lower factorials is given in **table 1**. Note that 0! is listed as 1. There is no logical reason for this, as 0! is a meaningless quantity, but by convention it is given the value of unity and by so doing many formulae using factorials can be simplified. However, this is a subject which needn't concern us here.

Returning to our corflake competition, the organisers are seldom generous enough to list only four features. A more usual number would be 8. Reference to the table tells us that we would need to submit 40,320 entries to be certain of getting that winning line. If the number was increased to 12 (as was the case with a recent national competition) over 479 *million* entries would be needed - and they even asked for a tiebreaker! (The exclamation mark at the end of the last sentence was, in this case, not intended as a factorial symbol.) Clearly, it can be seen that as the factorials increase they get radidly larger. For example, 52! - the number of different ways that a pack of cards can be arranged - is a 68-digit number. Look up factorial 52 in a book of mathematical tables and you will generally

```
0!=1
1!=1
2!=2
3!=6
4!=24
5!=120
6!=720
7!=5040
8!=40320
9!=362880
10!=3628800
11!=39916800
12!=479001600
13!=6227020800
14!=87178291200
15!=1307674368000
16!=20922789888000
17!=355687428096000
18!=6402373705728000
19!=121645100408832000
20!=2432902008176640000
21!=51090942171709440000
22!=1124000727777607680000
23!=25852016738884976640000
24!=620448401733239439360000
25!=15511210043330985984000000
26!=403291461126605635584000000
27!=10888869450418352160768000000
28!=304888344611713860501504000000
29!=8841761993739701954543616000000
30!=265252859812191058636308480000000

52!=80658175170943878571660636856403766
   -975289505440883277824000000000000
```

```
193227957864970790705422033741680812825210880916326
4166718699849528031656883994545647203712674067525
7118447137275484793742741837761505519594416711842
787117869763898069909087114888078977412488043866085
2219292529734883897107278374270145635533407357337
133106739557770672012016453554977614856667399969369
5139916953057024373220777072120066417985169160642 8
3594768604947699979669307180211118280732359229779047
15630052705144935277545642535797189115758007355607
947467040730021448675709445216418097801900129011 63
79192266129546927861227170828621497514515563075715
046276400089105833734683287831937049584691922249765
59253878136975399025116889109803013271740208004737
8736738468576314422838600616318236157511556858852
0269166527122903170647363721761172881046107338014
400507367403456506220079579922384113055897342056448
27574631080375706893155662910797243137953562170462
7688158075070860605800811139593214253177051202110
897129999586424264842748987879307712921882641033474
11893295122215091365752223548750234512052612897239
673552561097493779543177727382382050270408414397872
2247421373851145283875835953671620730818135509289
0575874133097833240202248705719777503377908992831332
02413020647274709457403181333292160521309544498420
48425770039582452887808840187722831622816731401648
99766900193709909232267375634735212603682490272891
2631085745363860501990127371682673294647052868967
3352612812371614237616943945060690427042419243345
84009318052174939581550531332884721946680679475983
7220889233778107323421287828379641252562301070211
69556126165448028195601177056733065329774518863955
78642028723425719834746668415713689784598037540393
40888032227708512724937579284430207811803134787 3064
45169085471302647598037213574317656261028080181477
671182272383551082692459160666920334294997958973100
63618310704468931265944839607660564097790328559514
91492322247132755696697703785909223753920436211973
5602745268066609383939577704630936702531604813486
68376070452821420166622765313911884306143974455165
549527698014338973827222155708849153563479874306101
03191228834169724464278834756173824589706977415525 9
51876485234878934721809617709760157238138221029818
2950898620953011371037197201155737260281385992002
97701526888443183322366091195977603488945641781764
8681837493120000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000
0000000000000000000000000000000000000000000000000
```

The 2401 digits in factorial 244

Prize

WELL, here we are, saved from plumbing the depths (and we do mean depths) of the Magic Bottomless Box by John Foster of **Kouga Software**, who wrote today to say that, having promised us five copies of the greatly lauded **Mandragore**, he reckoned he could manage another five. And we didn't even have to break his arms. Come to think of it, just as well we didn't, or he wouldn't have been able to write, would he? The anteater-infested arcade challenge has been pronounced the best game of 1988 by many, and is a generous contribution from a software company in its early days.

Rules

When you have cracked the 'cornflake' competition, packet into an envelope marked JANUARY COMPETITION with your answer, listing and any comments you care to add, and send it to us at Alexandra Road. No cassettes, please, unless they contain rare recordings of Led Zeppelin. Then wait for the crunch to come.

Talking of crunches, what about the tiebreaker? Ah yes! Using your skill and judgement, place five things you like about *Dragon User* in order of importance. Think carefully before answering this. You may enter as many times as you like.

October winners

We allowed a certain amount of flexibility in the final solution, as many entries came up with slightly different answers by perfectly fair means, However, it wasn't difficult to pick out the most confident calculations.

The winners are: E.A.Newman of Addlestone with a particularly far-reaching set of comments, S A Siddiqui of Chiswick, D J Gray of Middlesborough, Patricia Hill of Cashalton Beeches, P D Maddocks of Taplow, Peter Duncombe of Harpenden, Paul Weedon of Wotton-under-Edge, Fred Willers of Yarnfield and last but not least Austan Henderson of Bromsgrove, who beat several other entries of comparable brilliance on the strength of his tiebreaker. There you are - they even work down to 10th place. (The cast does not necessarily appear in order of success, but the battle for 10th place was definitely joined here.)

Austan's tiebreaker involves liking Indoor Football because it doesn't make him out of breath. We had a few this month ... I like indoor football better than outdoor football because the trainer doesn't use a nasty cold sponge for fear of spoiling the carpet ... because outdoors I get the feeling I'm playing the most gifted rugby players in the country ... because getting muddy is a Drag on my social life ... because it's difficult to plug a Dragon in on an outdoor football pitch (good one, that. Fundamentally true on all counts) ... because I won't have to go to Coventry to win my Spurs ... and so on. All good stuff.

Solution

See opposite.

find the value given as 8.06581 (67). (The number in brackets meaning that the decimal point needs to be moved 67 places to the right to obtain a value of the correct magnitude. Of course, this will not be the true value accurate to the last digit, but it will be sufficiently close for most practical purposes.

The calculation of such high factorials is generally restricted to the researches of the numerologist. However, a number of oddities have been found which relate to factorials. In 1876 a Frenchman, H. Brocard, noted that 4!, 5! and 7!, when increased by 1, became perfect squares. Using the methods of calculation then available he was unable to find others with this property, and so he conjectured that these were the only ones. Now, a century on, computers have taken the calculation of factorials far beyond any that Brocard would have considered possible - still without finding any more to add to the list. The value of 7! is interesting in that it has a square number of digits and can thus be printed in square formation. Other factorials with a square number of digits are the factorials of 12, 18, 32, 59 and 81. Of all values under 1000!, there are just 20 that can be printed in this way, the highest of them being 944! which has 2401 digits. For the numerically curious, this value is given here.

Another problem relating to factorials involves finding numbers in which the factorials of each individual digit adds up to the number itself. Apart from the trivial solutions of 1! and 2! there are just two possible numbers. One of these is 145, since 1! + 4! + 5! also equals 145. I will leave it to interested readers to calculate the other value (using a short computer program if necessary). However, there is a slight catch which has already been alluded to on this page!

This month's competition also involves factorials. Examination of the list of factorials in **table 1** reveals some oddities. Notice how the number of zeros at the end of each value gradually increases. These are cumulative and will increase without limit. For instance, the value of 944! shown here ends in 233 zeroes. Not so predictable, and hence more curious, are the repetitions within the factorials of other digits. Note the run of four consecutive 7s in 22! and the four 8s in 27! Other factorials with four repeating digits are 35!, 38!, 45! and 96! Even more unusual is the factorial of 151 which has six consecutive 3s amongst its 265 digits. Now, what we would like to know is the smallest factorial with *seven* digits all alike, and can you say what this digit is? Remember we are not considering any zeros (even if they occur away from the right-hand end of the value).

The Answer

This is Gordon Lee's own solution to the December competition
see page ★ ★ for results

THE Wallis and Leibniz formulae do not compute the fourth digit of pi until the 385th and 245th steps respectively.

Using the listing to generate Wallis' series, the approximations for the first few values for pi are given as:

4
2.66666667
3.55555556
2.84444444
3.41333333
2.92571429
3.34367347
2.97215420
3.30239355
3.00217596

```
   5 P$="3.141":Q=1:W=-1
  10 X=2:S=1:N=2:D=3
  20 Z=2*X
  30 GOSUB 1000
  40 X=X*N/D
  50 IF S/2=INT(S/2)THEN D=D+2 ELSE N=N+2
  60 S=S+1
  70 GOTO 20
1000 Z$=STR$(Z):Z$=MID$(Z$,2)
1010 U=VAL(MID$(Z$,Q,1))
1020 V=VAL(MID$(P$,Q,1))
1030 IF U<>V OR U<>W THEN W=U:GOTO 1060
1040 PRINT U;"  ";S-1
1050 W=-1:Q=Q+1:IF Q=2 THEN Q=3
1060 RETURN
```

From this it can be seen that the first digit - the 3 - is not computed definitely until the 9th step. That is, after the ninth step it *remains* as 3 and does not revert back to 2.

The listing given uses a subroutine which looks at the computed value of pi and compares the digit being tested with the actual value of the digit at that position in pi itself. Once the required digit has appeared in two consecutive assessments, the relevant values are printed out. The same routine can be used on both of the formulae as follows:
Add lines 5 which defines the extra variables.
Amend line 30 to read 30 GOSUB 1000
Add the subroutine (lines 1000 to 1060)

This is shown with the Wallis listing, but the procedure is exactly the same for the Leibniz formula.

Classified

Chip swap

Graham Smith provides 'Eprom Switching' in a DeltaDOS cartridge

THOSE of you with a DeltaDOS cartridge may have noticed that you have a spare socket sitting next to your DOS eprom. You may know that Premier produced a couple of utility Eproms to fit the memory area $E000-$FFE0 which is vacant above the DOS rom. The circuit layout is designed to enable one eprom socket when memory is accessed between $C000 and $DFFF, and the other eprom socket when memory is accessed between $E000 to $FFFF. This of course means that you can't swap the DOS chip and the *Encoder09* chip around because the data contained on them would find itself located in the wrong memory area. The important pin is the OUTPUT ENABLE (pin 22) on the 2764 Eprom (in fact pin 20 is the CHIP ENABLE but on both sockets pin 20 is strapped to pin 22).

Now, the point of this article is to explain how you can modify the circuitry so that the spare socket can be used to hold an alternative DOS (such as DOSplusDELTA), which would normally be a direct replacement for the existing DeltaDOS eprom and therefore expects to find itself in the socket tied to memory location $C000 to $DFFF. If you simply place it in the spare socket, it cannot function because it is the wrong memory area, however with an aid of a suitable switch, a few wires and a couple of cuts in the printed circuit, you can switch the sockets over. The advantage of this is that you can have both eproms permanently installed, eliminating the chances of damaging them when you need to change from one to the other. The disadvantage is the fact that you will lose the option to have the *Encoder09* or *Toolkit* chip fitted. You should also bear in mind that you should-not attempt to swap with the power on.

If you are confident of your soldering technique and are determined to have a go, then read on. Remember to read the following instructions carefully as you will have to cut some tracks on the board and solder onto the circuitry. If you get it wrong, it is your problem, I do not guarantee anything here. The principle is very simple, all you are doing is to install a switch which will toggle the OUTPUT ENABLE signal between the two sockets. You will need a double pole change-over switch, (I used a small sliding switch from Maplin), and four short lengths of wire (about 9 inches each should be enough). I suggest you use four different colours so you can keep track.

Before you start, decide where you are going to mount the switch on the cartridge case and ensure that it will not foul any of the components and that you will be able to get the cartridge back together. Also make sure that you can still insert the cartridge in the side of the Dragon and that you can still fit the DOS connecting socket in place. Make sure that the lengths of wire that you are going to use, will reach from where they are to be terminated, to the final location of the switch. These may seem obvious, but if you forget them, you will be sorry.

First, find chip IC7. There is a small indentation at one end to indicate the top. If you look at the chip so that the indentation is at the top, pin 1 is the top left, pin 7 is the bottom left, pin 8 is the bottom right, pin 14 is the top right. In other words, the pins number anti-clockwise around the chip. Count around carefully to pin 11 (this should be the middle pin on the right hand side). You will see a printed circuit track leaving the base of this pin, (note pin 11). Cut this track. Be very careful not to cut any of the other tracks near it. I made two cuts close together across the track and scraped the track from between them (less than a millimeter). Remember the track is only on the surface of the board, you don't have to go in deep.

Second, find three little holes marked LK2 (A,B,C) and trace the track leaving LK2 'A'. Follow the track and you will find that it eventually ends on LK3 'C'. Cut this track just outside the white box surrounding the LK2 holes (note LK2 holes).

Third, there may be a wire link between LK3 'B' and 'C'. If there is, remove it (unsolder it). Fit a wire link between LK3 'A' and 'B'.

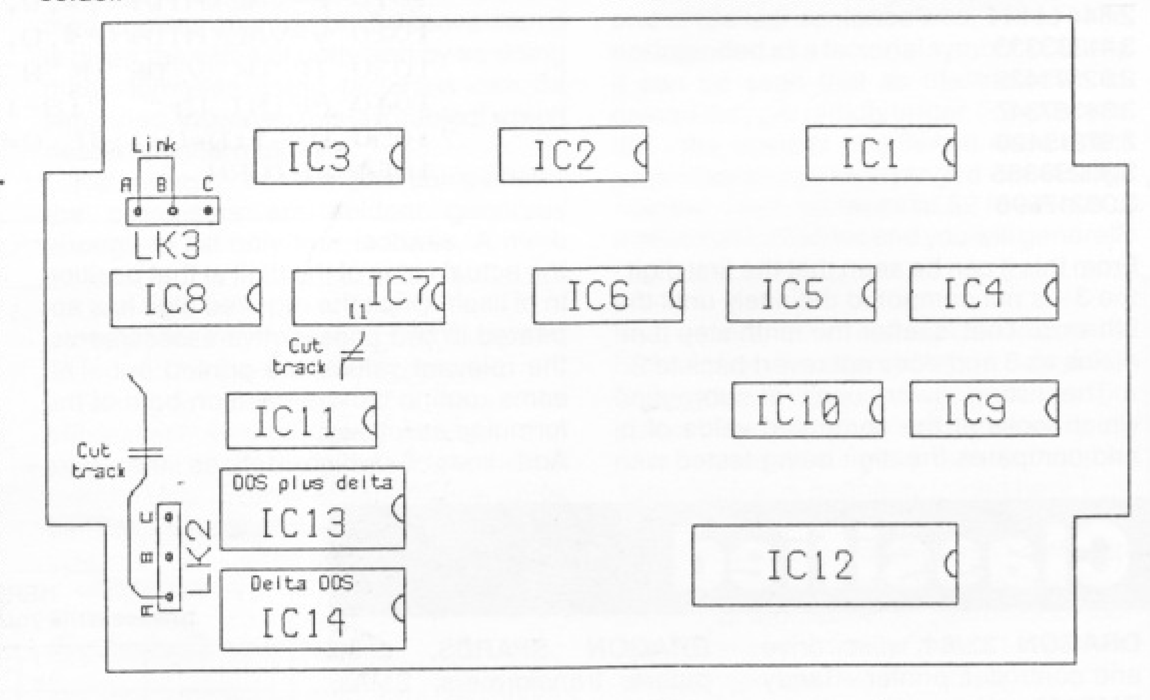
Fourth, let's define the contracts on your switch. It should have six terminals (legs), in two rows of three. Let's call one side 1, 2, 3 and the other side 4, 5, 6 counting anti-clockwise. You now have some wires to solder.

1. Connect switch leg 1 to switch leg 4
2. Connect switch leg 3 to switch leg 6
3. Connect switch leg 1 to LK2 'A'
4. Connect switch leg 3 to IC13 pin 22
5. Connect switch leg 2 to LK3 'C'
6. Connect switch leg 5 to IC7 pin 11

Now read through those steps again and see if you did it right.

Finally, insert your alternative DOS chip in the spare socket. Put the cartridge back together and connect up the drive lead. Insert the cartridge and power up. Depending on which way you left the switch, you will either get the DeltaDOS screen or your alternative DOS screen. Power off, slide the switch to the other position, power on and hopefully you should get the other DOS screen. The first time I tried this modification, nothing would work. The problem was simply the fact that the switch was faulty, and nothing to do with the modifications, but it did give me a nasty few moments. I had actually purchased three switches because they were cheap and on testing I found that only one was reliable. Obviously the moral is, you get what you pay for.

If the idea of an alternative DOS for your Delta cartridge catches your imagination, I will just add that DOSplusDELTA is the only alternative DOS I have heard of for Delta users. It is available from me at Orange Software or from Phil Scott direct.



Delta DOS cartridge layout

